64 kbps DSO. Despite these comparisons the question remains.

Although routers are much cheaper than telephone switches, they have much less capacity.

Building large networks with small building blocks gets not only expensive, but quickly reaches points of diminishing returns. We already have seen the Internet backbone get overloaded with the current crop of high end routers, and they are yet to experience the significant traffic increase that a successful Internet Telephony offering would bring. We are saying two things here.

1. It is unlikely that the current Internet backbone can support a major traffic increase associated with a successful internet telephony service. We need to wait for the technology of routers to improve.

2. The second issue raised above was that of bandwidth usage. Indeed 10 kbps half duplex (a little more when both parties occasionally speak at the same time, but much less during periods of silence) is considerably less than 64 kbps full duplex dedicated capacity.

Two points should be noted on this argument.

First, bandwidth is cheap, at least, when there is spare fiber in the ground. Once the last strand is used the next bit per second is very expensive. Second, on transoceanic routes, where bandwidth is much more expensive, we are already doing bandwidth compression of voice to 9.6 kbps. This is essentially equivalent to the 10 kbps of Internet Telephony.

Why is IP capacity priced so much cheaper than POTS? The answer is that the pricing difference is partly related to the subsidized history of the Internet. There is a process in motion today, by the Internet backbone providers, to address some of the cost issues of the Internet. The essence of the process is the recognition that the Internet requires a usage charge. Such charges already apply to some dial up users, but typically do not apply to users with dedicated connections.

If PC to PC Internet Telephony becomes popular, users will tend to keep their PCs connected for long periods. This will make them available to receive calls. It will also drive up hold times on dial in ports. This will have a significant effect on the capital and recurring costs of the Internet.

(5) Charges A directory service must provide the functions described above and collect enough information to bill for the service. A charge can be made for directory service as well as for registration (a one time fee plus a monthly fee), call setup, but probably not for duration.

Duration is already charged for the Internet dial in user and is somewhat bundled for the LAN-attached user. Usage charges for Internet service may be coming soon (as discussed above). Duration charges are possible for the incoming and outgoing PSTN segments.

Incoming PSTN calls may be charged as the long distance segment by using a special area code. Other direct billing options are 800 calls and calling card (or credit card) billing options (both require a second dial tone).

Requiring all callers (except incoming PSTN calls) to be registered with the directory service will eliminate the immediate need for most collect calling. This will probably not be a great impediment since most users of the IP Phone service will want to receive as well as originate calls, and registration is required for receiving calls. Callers could have unlisted entries which would be entries with an E.164 address, but no name. People given this E.164 address could call the party (from the PSTN or from a PC), as is the case in the present phone system.

Different compression levels can be used to provide different quality of voice reproduction and at the same time use more or less Internet transit resources. For PC to PC connections the software packages at both ends can negotiate the amount of bandwidth to be used. This negotiation might be facilitated through the directory service.

(6) Technical Issues It will be necessary to coordinate with IP Phone vendors to implement the registration.

automatic presence notification, and verification capabilities. We will also need to add the ability to communicate service requests. These will include authorization for collect calls specifying attributes such as "place a dial out call to the PSTN even if it is long distance" and others to be determined.

Registration with a directory is a required feature that will be illuminated below. Using the DNS model for the distributed directory service will likely facilitate this future requirement.

Assignment of a pseudo E. 164 number to directory entries will work best if a real area code is used. If each carrier has an area code it will make interworking between the directory systems much easier. An obvious complication will arise when number portability becomes required.

IP Telephony, in accordance with a preferred embodiment, is here and will stay for at least the near future. A combination of a carrier level service, based on this technology, and a growth in the capacity of routers may lead to the internet carrying a very significant percentage of future long distance traffic.

The availability of higher speed Internet access from homes, such as cable modems, will make good quality consumer IP Telephony service more easily attained. The addition of video will further advance the desirability of the service.

More mundane, but of interest, is FAX services across the Internet. This is very similar to the voice service discussed above. Timing issues related to FAX protocols make this a more difficult offering in some ways.

Conferencing using digital bridges in the Internet make voice and video services even more attractive. This can be done by taking advantage of the multi-casting technology developed in the Internet world. With multi-casting the cost of providing such services will be reduced.

C. Internet Telephony Services Figure 1C is a block diagram of an internet telephony system in accordance with a preferred embodiment. Processing commences when telephone 200 is utilized to initiate a call by going off hook when a party dials a telephone number. Telephone 200 is typically connected via a conventional two-wire subscriber loop through which analog voice signals are conducted in both directions. One of ordinary skill in the art will readily realize that a phone can be connected via fiber, ISDN or other means without departing from the teaching of the invention. Alternatively, a person could dial a phone number from a computer 210, paging system, video conferencing system or other telephony capable devices. The call enters a Local Exchange Carrier (LEC) 220 which is another name for a Regional Bell Operating Company (RBOC) central switch. The call is terminated by a LEC at a leased Common Business Line (CBL) 230 of an interchange carrier such as MCI. As a result of the termination to the CBL, the MCI Switch 221 receives an offhook indication.

The Switch 221 responds to the offhook by initiating a DAL Hotline procedure request to the Network Control System (NCS) which is also referred to as a Data Access Point (DAP) 240.

The switch 221 is simplified to show it operating on a single DS1 line, but it will be understood that switching among many lines actually occurs so that calls on thousands of individual subscriber lines can be routed through the switch on their way to ultimate destinations. The DAP 240 returns a routing response to the originating switch 221 which instructs the originating switch 221 to route the call to the destination switch 230 or 231. The routing of the call is performed by the DAP 240 translating the transaction information into a specific SWitch ID (SWID) and a specific Terminating Trunk Group (TTG) that corresponds to the route out of the MCI network necessary to arrive at the appropriate destination, in this case either switch 230 or 231. An alternative embodiment of the hybrid network access incorporates the internet access facility into a switch 232. This integrated solution allows the switch 232 to attach directly to the internet 295 which reduces the number of network ports necessary to connect the network to the internet 295. The DAP sends this response

information to the originating switch 221 which routes the original call to the correct Terminating Switch 230 or 231. The terminating switch 230 or 231then finds the correct Terminating Trunk Group (TTG) as indicated in the original DAP response and routes the call to the ISN 250 or directly to the modem pool 270 based on the routing information from the DAP 240. If the call were destined for the Intelligent Services Network (ISN) 250, the DAP 240 would instruct the switch to terminate at switch 230.

Based upon analysis of the dialed digits, the ISN routes the call to an Audio Response Unit (ARU) 252. The ARU 252 differentiates voice, fax, and modem calls. If the call is a from a modem, then the call is routed to a modem pool 271 for interfacing to an authentication server 291to authenticate the user. If the call is authenticated, then the call is forwarded through the UDP/IP or TCP/IP LAN 281 or other media communication network to the Basic Internet Protocol Platform (BIPP) 296 for further processing and ultimate delivery to a computer or other media capable device.

If the call is voice, then the ARU prompts the caller for a card number and a terminating number. The card number is validated using a card validation database. Assuming the card number is valid, then if the terminating number is in the US (domestic), then the call would be routed over the current MCI voice lines as it is today. If the terminating number is international, then the call is routed to a CODEC 260 that converts the voice to TCP/IP or UDP/IP and sends it via the LAN 280 to the internet 295. The call is routed through a gateway at the terminating end and ultimately to a phone or other telephony capable device.

Figure 1D is a block diagram of a hybrid switch in accordance with a preferred embodiment.

Reference numbers have been conserved from Figure 1C, and an additional block 233 has been added. Block 233 contains the connecting apparatus for attaching the switch directly to the internet or other communication means. The details of the connecting apparatus are presented in Figure 1E. The principal difference between the hybrid switch of Figure 1D and the switches presented in Figure 1C is the capability of switch 221 attaching directly to the Internet 295.

Figure 1E is a block diagram of the connecting apparatus 233 illustrated in Figure 1D in accordance with a preferred embodiment. A message bus 234 connects the switch fabric to

an internal network 236 and 237. The internal network in turn receives input from a Dynamic Telephony Connection (DTC) 238 and 239 which in turn provides demuxing for signals originating from a plurality of DS1 lines 242, 243, 244 and 245. DS1 lines, described previously, refer to the conventional bit format on the TI lines.

To accommodate the rapidly diversifying telephony / media environment, a preferred embodiment utilizes a separate switch connection for the other internal network 237. A Spectrum Peripheral Module (SPM) 247 is utilized to handle telephony/media signals received from a pooled switch matrix 248, 249, 251, 254, 261-268. The pooled switch matrix is managed by the SPM 247 through switch commands through control lines. The SPM 247 is in communication with the service provider's call processing system which determines which of the lines require which type of hybrid switch processing. For example, fax transmissions generate a tone which identifies the transmission as digital data rather than digitized voice. Upon detecting a digital data transmission, the call processing system directs the call circuitry to allow the particular input line to connect through the pooled switch matrix to a corresponding line with the appropriate processing characteristics. Thus, for example, an internet connection would be connected to a TCP/IP Modem line 268 to assure proper processing of the signal before it was passed on through the internal network 237 through the message bus 234 to the originating switch 221 of Figure 1D.

Besides facilitating direct connection of a switch to the internet, the pooled switch matrix also increases the flexibility of the switch for accommodating current communication protocols and future communication protocols. Echo cancellation means 261 is efficiently architected into the switch in a manner which permits echo cancellation on an as-needed basis. A relatively small number of echo cancellers can effectively service a relatively large number of individual transmission lines. The pooled switch matrix can be configured to dynamically route either access-side transmissions or network-side transmissions to OC3 demux, DSP processing or other specialized processing emanating from either direction of the switch.

Moreover, a preferred embodiment as shown in Figure 1E provides additional system efficiencies such as combining multiplexer stages in a port device on one side of a voice or data circuit switch to enable direct connection of a fiber-optic cable to the multiplexed output of the port device. Moreover, redundancy is architected into the switch through the alternate

routes available over CEM 248 / 249 and RM 251 254 to alternate paths for attaching various communication ports.

When the switch 221 of Figure 1D, is connected to the internet 295, the processing is provided as follows. A line from the internet 295 enters the switch through a modem port 268 and enters the pooled switch matrix where demux and other necessary operations are performed before the information is passed to the switch 221 through the internal network 237 and the message bus 234. The modules 261-268 provide plug and play capability for attaching peripherals from various communication disciplines.

Figure 1F is a block diagram of a hybrid (internet-telephony) switch in accordance with a preferred embodiment. The hybrid switch 221 switches circuits on a public switched telephone network (PSTN) 256 with TCP/IP or UDP/IP ports on an internet network 295.

The hybrid switch 221 is composed of PSTN network interfaces (247, 260), high-speed Internet network interfaces (271, 272, 274), a set of Digital Signal Processor (DSP)s (259, 263), a time-division multiplexed bus 262, and a high-speed data bus 275.

The hybrid internet telephony switch 221 grows out of the marriage of router architectures with circuit switching architectures. A call arriving on the PSTN interface 257 is initiated using ISDN User Part (ISUP) signaling, with an Initial Address Message (IAM), containing a called party number and optional calling party number. The PSTN interface 257 transfers the IAM to the host processor 270. The host processor

270 examines the PSTN network interface of origin, the called party number and other IAM parameters, and selects an outgoing network interface for the call. The selection of the outgoing network interface is made on the basis of routing tables. The switch 221 may also query an external Service Control Point (SCP) 276 on the internet to request routing instructions. Routing instructions, whether derived locally on the switch 221 or derived from the SCP 276, may be defined in terms of a subnet to use to reach a particular destination.

Like a router, each of the network interfaces in the switch 221 is labeled with a subnet address. Internet Protocol (IP) addresses contain the subnet address on which the computer is located. PSTN addresses do not contain IP subnet addresses, so subnets are mapped to PSTN area codes and exchanges. The switch 221 selects routes to IP addresses and PSTN addresses

by selecting an interface to a subnet which will take the packets closer to the destination subnet or local switch.

The call can egress the switch via another PSTN interface 258, or can egress the switch via a high-speed internet network interface 273. If the call egresses the switch via the PSTN interface 258, the call can egress as a standard PCM Audio call, or can egress the switch as a modem call carrying compressed digital audio.

In the case where the call egresses the switch 221 as a standard PCM audio call, the PCM audio is switched from PSTN interface 257 to PSTN interface 258 using the TDM bus 260.

Similarly, PCM audio is switched from PSTN Interface 258 to PSTN interface 257 using the TDM bus 260.

In the case where the call egresses the switch 221 as a modem call carrying compressed digital audio, the switch 221 can initiate an outbound call to a PSTN number through a PSTN interface 258, and attach across the TDM Bus 260 a DSP resource 259 acting as a modem.

Once a modem session is established with the destination, the incoming PCM audio on PSTN interface 257 can be attached to a DSP Resource 263 acting as an audio codec to compress the audio. Example audio formats include ITU G.729 and G.723. The compressed audio is packetized into Point to Point Protocol (PPP) packets on the DSP 263, and transferred to DSP 259 for modem delivery over the PSTN Interface 258.

In the case where the call egresses the switch 221 on a high speed internet interface 272, the switch 221 attaches the PSTN interface 257 to the DSP resource 263 acting as an audio codec to compress the PCM audio, and packetize the audio into UDP/IP packets for transmission over the Internet network. The UDP/IP packets are transferred from the DSP resource 263 over the high-speed data bus 275 to the high-speed internet network interface 272.

Figure 1G is a block diagram showing the software processes involved in the hybrid internet telephony switch 221. Packets received on the internet network interface 296 are transferred to the packet classifier 293. The packet classifier 293 determines whether the packet is a normal IP packet, or is part of a routing protocol (ARP, RARP, RIP, OSPF, BGP, CIDR) or management protocol (ICMP). Routing and management protocol packets are handed off to

the Routing Daemon 294. The Routing Daemon 294 maintains routing tables for the use of the packet classifier 293 and packet scheduler 298. Packets classified as normal IP packets are transferred either to the packetizer/depacketizer 292 or to the packet scheduler 298.

Packets to be converted to PCM audio are transferred to the packetizer/depacketizer 292.

The packetizer/depacketizer takes packet contents and hands them to the codec 291, which converts compressed audio into PCM Audio, then transfers PCM audio to the PSTN Interface 290.

Normal IP packets to be sent to other internet devices are handed by the packet classifier 293 to the packet scheduler 298, which selects the outgoing network interface for the packet based on the routing tables. The packets are placed upon an outbound packet queue for the selected outgoing network interface, and the packets are transferred to the high speed network interface 296 for deliver across the internet 295.

D. Call Processing This section describes how calls are processed in the context of the networks described above.

1. VNET Call Processing Figure 10A illustrates a Public Switched Network (PSTN) 1000 comprising a

local exchange (LEC) 1020 through which a calling party uses a telephone 1021 or computer 1030 to gain access to a switched network including a plurality of MCI switches 1011, 1010. Directory services for routing telephone calls and other information is provided by the directory services 1031 which is shared between the Public Branch Exchanges 1041, 1040 and the PSTN.

This set of scenarios allows a subscriber to use either a PC, telephone or both to make or receive VNET calls. In this service, the subscriber may have the following equipment: A telephone that uses VNET routing is available today in MCI's network. In this case, VNET calls arriving in the MCI PSTN network using the subscriber's VNET number are routed with the assistance of the DAP just as they are routed today.

A PC that is capable of Internet telephony. Calls are routed into and out of this PC with the assistance of an Internet or Intranet Directory Service that tracks the logged-in status and current IP address of the VNET user.

A PC and a telephone is used to receive and make calls. In this case, a user profile will contain information that allows the DAP and Directory Service to make a determination whether to send an incoming call to the PC or to the telephone. For example, the user may always want calls to go to their PC when they are logged-in and to their phone at all other times. Or, they may want their calls to always go to their PC during normal work hours and to their phone at other times. This type of control over the decision to send incoming calls to a phone or PC may be controlled by the subscriber.

The following scenarios apply to this type of service.

1. A PC to PC call where the Directory service is queried for the location of the terminating PC. PCs connected to an intranet using the Intranet as transport.

Both PC's connected to a corporate Intranet via dial up access.

Both PCs on separate Intranets with the connection made through the Internet.

Both PCs on the Internet through a dial-up connection.

One PC directly connected to a corporate Intranet and the other PC using a dial-up connection to the Internet.

One PC using a dial up connection to a corporate Intranet and the other PC using a dial- up connection to the Internet.

Both PCs on separate Intranets with the connection made through the PSIN.

One or both PCs connected to a corporate Intranet using dial-up access.

One or both of the PCs connected to an Internet Service Provider.

One or both of the ITGs as an in-network element.

2. A PC to phone call where a directory service is queried to determine that the terminating VNET is a phone. The PC then contacts an Internet Telephony Gateway to place a call to the terminating phone.

PC on an intranet using a private ITG connected to the PSTN with the ITG as an out of network element. The destination phone is connected to a PBX.

The PC may also be using a public ITG that must be access through the Internet.

The PC may be connected to the corporate Intranet using dial-up access.

PC on an intranet using a private ITG connected to the PSTN with the ITG as an in- network element. The destination phone is connected to a PBX.

The PC may also be using a public ITG that must be accessed through the Internet.

The PC may be connected to the corporate Intranet using dial-up access.

PC on an intranet using a private ITG connected to the PSTN with the ITG as an in- network element. The destination phone is connected to the PSTN.

The PC may also be using a public ITG that must be accessed through the Internet.

The PC may be connected to the corporate Intranet using dial-up access.

The ITG may be an in-network element.

PC on an intranet using a private ITG connected to a PBX with the traffic carried over the Intranet.

PC is at a different site than the destination phone with the traffic carried over the internet or intranet.

The PC may be using a dial-up connection to the corporate Intranet.

3. A phone to PC call where the DAP or PBX triggers out to the Internet Directory Service to identify the terminating IP address and ITG for routing the call. The call is then routed through the PSTN to an ITG and a connection is made from the ITG to the destination PC.

Possible Variations: Same variations as the PC to phone.

4. A Phone to Phone call where the DAP or PBX must query the Directory Service to determine whether the call should be terminated to the subscriber's phone or PC.

Possible Variations: Both Phones are on a PBX;

One phone is on a PBX and the other phone is on the PSTN: and Both phones are on the PSTN.

For each of these variations, the DAP and Directory Service may be a single entity or they may be separate entities. Also, the directory service may be a private service or it may be a shared service. Each of the scenarios will be discussed below with reference to a call flow description in accordance with a preferred embodiment. A description of the block elements associated with each of the call flow diagrams is presented below to assist in understanding the embodiments.

2. Descriptions of Block Elements Element Description Phl | Traditional analog phone connected to a Local Exchange Carrier. For the purposes of or these VNET scenarios, the phone is capable of making VNET calls, local calls or DDD calls. In some scenarios the VNET access may be done through The customer dials a 700 number with the last seven digits being the destination VNET number for the call. The LEC will know that the phone is picked to MCI and route the call to the MCI switch. The MCI switch will strip off the "700", perform and ANI lookup to identify the customer ID and perform VNET routing using the VNET number and customer ID. The customer dials an 800 number and is prompted to enter their Social Security number (or other unique id) and a VNET number. The switch passes this infonnation to the DAP which does the VNET translation. PCI Personal computer that has the capability to dial in to an Internet service provider PC2 or a corporate intranet for the purpose of making or receiving internet telephony calls. The following access methods might be used for this PC Internet service provider The PC dials an 800 number (or any other dial plan) associated with the service provider and is routed via normal routing to the modem bank for that provider. The user of the PC then follows normal log-on procedures to connect to the internet. Corporate Intranet The PC dials an 800 number (or any other dial plan) associated with the corporate Intranet and is routed via normal routing to the modem bank for that Intranet. The user of the PC then follows normal log-on procedures to connect to the Intranet. LEC SFI Switching fabric for a local exchange carrier. This fabric provides the connection between Phl/PCI/PC2 and MCI's telephone network. It also provides local access to customer PBXs. MCI SFI Switching fabric for MCI (or for the purpose of patenting, any telephony service MCI SF2 provider). These SFs are capable of performing traditional switching capabilities for MCI's network. They are able to make use of advanced routing capabilities such as those found in MCI's NCS (Network Control System). NCS The NCS provides enhanced routing services for MCI. Some of the products that are supported on this platform are: 800, EVS, Universal Freephone, Plus Freephone, Inbound International, SAC(ISAC) Codes, Paid 800, 8XX/Vnet Meet Me Conference Call, 900, 700. PCS, Vnet, Remote Access to Vnet, Vnet Phone Home, CVNS, Vnet Card, MCI Card (950 Cards), Credit Card and GETS Card. In support of the existing VNET services, the DAP provides private dialing plan capabilities to Vnet customers to give them a virtual private network. The DAP supports digit translation, origination screening, supplemental code screening, 800 remote access, and some special features such as network call redirect for this service. To support the call scenarios in this document, the NCS also has the capability to made a data query to directory services in order to route calls to PCs. Dir Svc 1 Internet Directory Services. The directory service performs: Dir Svc 2 Call routing - As calls are made to subscribers using internet telephony services from MCI, the directory service must be queried to determine where the call should terminate. This may be done based upon factors such as - the logged-in status of the subscriber. - service subscriptions identifying the subscriber as a PC or phone only user - preferred routing choices such as "route to my PC always if I am logged in", or route to my PC from 8-5 on weekdays, phone all other times". etc. Customer profile management - The directory service must maintain a profile for each subscriber to be able to match VNET numbers to the service subscription and current state of subscribers. Service authorization - As subscribers connect their PCs to an IP telephony service, they must be authorized for use of the

service and may be given secunty tokens or encryption keys to ensure access to the service. This authorization responsibility might also place restrictions upon the types of service a user might be able to access, or introduce range privileges restncting the ability of the subscriber to place certain types of calls. ITG 1 Internet Telephony Gateway - The internet Telephony Gateway provides a path ITG 2 through which voice calls made be bridged between an IP network and a traditional telephone network. To make voice calls from an IP network to the PSTN, a PC software package is used to establish a connection with the ITG and request that the ITG dial out on the PSTN on behalf of the PC user. Once the ITG makes the connection through the voice network to the destination number, the ITG provides services to convert the IP packetized voice from the PC to voice over the PSIN. Similarly, the ITG will take the voice from the PSTN and convert it to IP packetized voice for the PC. To make voice calls from the PSTN to the IP network, a call will be routed to the ITG via PSTN routing mechanisms. Once the call arrives, the ITG identifies the IP address for the destination of the call, and establishes an IP telephony session with that destination. Once the connection has been established, the ITG provides conversion services between IP packetized voice and PCM voice. ITG 3 These ITGs act in a similar capacity as the ITGs connected to the PSTN. but ITG 4 these ITGs also provide a connection between the corporate Intranet and the PBX. IAD 1 The Internet access device provides general dial-up Internet access from a user's IAD 2 PC to the Internet. This method of connecting to the internet may bc used for Internet telephony, but it may also be simply used for Internet access. When this device is used for Internet telephony, it behaves differently than the ITG. Although the IAD is connected to the PSTN, the information traveling over that interface is not PCM voice, it is IP data packets. In the case of telephony over the IAD, the IP data packets happen to be voice packets, but the IAD has no visibility into those packets and cannot distinguish a voice packet from a data packet. The IAD can be thought of as a modem pool that provides access to the internet. PBX 1 Private Brach Exchange - This is customer premise equipment that provides PBX 2 connection between phones that are geographically co-located. The PBX also provides a method from those phones to make outgoing calls from the site onto he PSTN. Most PBXs have connections to the LEC for local calls, and a DAL connection to another service provider for VNET type calls. These PBXs also show a connection to a Directory Service for assistance with call routing. This capability does not exist in today's PBXs, but in the VNET call flows for this document, a possible interaction between the PBX and the Directory Service is shown. These PBXs also show a connection to an ITG. These ITGs provide the bridging service between a customer's Intranet and the traditional voice capabilities of the PBX. Phl 1 These are traditional PBX connected phones. Phl2 Ph21 Ph22 PC 11 These are customer premises PCs that are connected to customer Intranets. For the PC12 purposes ofthese call flows, the PCs have Internet Telephony software that allow

PC21 the user to make or receive calls. PC22

E. Re-usable Call Flow Blocks 1. VNET PC connects to a corporate intranet and logs in to a directory service Direfory I Services VNE 'Passwor4IP. I)PCOnline *coai.. uthenticate user I (p 2) PC Online Acli r Ack, Issecunt: alpdate Pfile wU VNET PC connects i ndConfigdata to corporate Intranet * Optional data depending upon implementation | [* Optional data depending upon implementation 1. The user for a PC connects their computer to an IP network, turns on the computer and starts an IP telephony software package. The software package sends a message to a directory service to register the computer as "on-line" and available to receive calls. This on-line registration message would most likely be sent to the directory service in an encrypted format for security. The encryption would be based upon an common key shared between the PC and the directory service. This message contains the following information: Some sort of identification of the computer or virtual private network number that may be used to address this computer. In this VNET scenario, this is the VNET number assigned to the individual using this PC. This information will be used to identify the customer profile associated with this user. It may also be some identification such as name, employee id, or any unique ID which the directory service can associate with a VNET customer profile.

. A password or some other mechanism for authenticating the user identified by the VNET number.

The IP address identifying the port that is being used to connect this computer to the network. This address will be used by other IP telephony software packages to establish a connection to this computer.

The message may contain additional information about the specifics of the software package or PC being used for IP telephony and the configuration/capabilities of the software or PC. As an example it might be important for a calling PC to know what type of compression algorithms are being used, or other capabilities of the software or hardware that might affect the ability of other users to connect to them or use special features during a connection.

The location of the directory service to receive this "on-line" message will be determined by the data distribution implementation for this customer. In some cases this may be a private database for a company or organization subscribing to a VNET service, in other cases it might be a national or worldwide database for all customers of a service provider (MCI). This location is configured in the telephony software package running on the PC.

2. When the directory service receives this message from the PC, it validates the user by using the VNET number to look up a user profile and comparing the password in the profile to the password received. Once the user has been validated, the directory service will update the profile entry associated with the VNET number (or other unique ID) to indicate that the user is 'on-line" and is located at the specified IP address. The directory service will also update the profile with the configuration data sent during the login request. Upon successful update of the, the directory service sends a response back to the specified IP address indicating that the message was received and processed. This acknowledgment message may also contain some sort of security or encryption key to guarantee secure communication with the directory service when issuing additional commands. When the PC receives this response message it may choose to notify the user via a visual or audible indicator.

Variation for On-line registration The call flow segment shown earlier in this section showed a PC on-line registration where the PC simply sends a password to the directory service to log-on. A variation for this log-on procedure would be the following call flow segment where the directory

service presents a challenge and the PC user must respond to the challenge to complete the log-in sequence. This variation on the log-in sequence is not shown in any of the call flows contained within this document, but it could be used in any of them. pe Directory Services VNET.IP. I)PCOnlinea Cahllate 2) Directory Servee Calculate'1 ChallenFe | I) PC Online r | I calclliate | Challenge Challenge 1 2) DirectoryService I< v J Challenge Calculate 1 3)ChallengeResponse ReSpOnse vL Response I I T ~ T 7 Authenticate user Ack Challenge Response I Update Profile with IP 4) PC Online Ack and Ack. Secllrily liey t W Config data * Optional data depending upon implementation 1. The user for a PC connects their computer to an IP network, turns on the computer and starts an IP telephony software package. The software package sends a message to a directory service to register the computer as "on-line" and available to receive calls. This on-line registration message would most likely be sent to the directory service in an encrypted format for security. The encryption would be based upon an common key shared between the PC and the directory service. This message contains the following information: Some sort of identification of the computer or virtual private network number that may be used to address this computer. In this VNET scenario, this is the VNET number assigned to the individual using this PC. This information will be used to identify the customer profile associated with this user. It may also be some identification such as name, employee id, or any unique ID which the directory service can associate with a VNET customer profile.

The IP address identifying the port that is being used to connect this computer to the network. This address will be used by other IP telephony software packages to establish a connection to this computer.

The message may contain additional information about the specifics of the software package or PC being used for IP telephony and the configuration/capabilities of the software or PC. As an example it might be important for a calling PC to know what type of compression algorithms are being used, or other capabilities of the software or hardware that might

affect the ability of other users to connect to them or use special features during a connection.

The location of the directory service to receive this 'on-line" message will be determined by the data distribution. implementation for this customer. In some cases this may be a private database for a company or organization subscribing to a VNET service, in other cases it might be a national or worldwide database for all customers of a service provider (MCI). This location is configured in the telephony software package running on the PC.

2. In this scenario the PC did not provide a password in the initial registration message. This is because the directory service uses a challenge/response registration process. In this case, the directory service will use a shared key to calculate a challenge that will be presented to the PC 3. The PC receives this challenge and presents it to the user of the PC. The PC user uses the shared key to calculate a response to the challenge and send the response back to the directory service

4. When the directory service receives this response from the PC, it validates the user. Once the user has been validated, the directory service will update the profile entry associated with the VNET number (or other unique ID) to indicate that the user is 'on-line" and is located at the specified IP address. The directory service will also update the profile with the configuration data sent during the login request. Upon successful update of the, the directory service sends a response back to the specified IP address indicating that the message was received and processed. This acknowledgment message may also

contain some sort of security or encryption key to guarantee secure communication with the directory service when issuing additional commands. When the PC receives this response message it may choose to notify the user via a visual or audible indicator.

2. VNET PC queries a directory service for a VNET translation PC Directors Service Source VNET. IP. Dest VNET'Config Data I) VNET Translation Req Match VNETTranslationReq I lntch VNET to profile Detennine rolite Check coizfigtiratioll \) 2) VNET Tmualatiau Reap IP. 'Config Data PC translation or IF. Dialed Number * Optional data depending upon implementation 1. A PC uses an Internet telephony software package to attempt to connect to a VNET number. To establish this connection, the user of the PC dials the VNET number (or other unique ID such as name, employee ID, etc). Once the telephony software package has identified this call as a VNET type call , it will send a translation request to the directory service. At a minimum, this translation request will contain the following information: The IP address of the computer sending this request The VNET number of the PC sending this request.

The Vnet number (or other ID) of the computer to be dialed.

A requested configuration for the connection. For example, the calling PC might want to use white-board capabilities within the telephony software package and may wish to verify this capability on the destination PC before establishing a connection. If the VNET number does not translate to a PC, this configuration information may not provide any benefit, but at the time of sending this request the user does not know whether the VNET number will translate to a PC or phone.

2. When the directory service receives this message, it uses the Vnet number (or other ID) to determine if the user associated with that VNET number (or other ID) is "on-line" and to identify the IP address of the location where the computer may be contacted.

This directory service may also contain and make use of features like time of day routing, day of week routing, ANI screening, etc.

If the VNET number translates into a PC that is "on-line", the directory service will compare the configuration information in this request to the configuration information

available in the profile for the destination PC. When the directory service returns the response to the translation request from the originating PC, the response will include The registered "on-line" IP address of the destination PC. This is the IP address that the originating PC may use to contact the destination PC Configuration information indicating the capabilities of the destination PC and maybe some information about which capabilities are compatible between the origination and destination PC.

If the VNET number translates to a number that must be dialed through the PSTN, the response message to the PC will contain the following - An IP address of an Internet Telephony gateway that may be used to get this call onto MCI's PSTN. The selection of this gateway may be based upon a number of selection algorithms. This association between the caller and the ITG to be used is made based upon information in the profile contained within the directory service.

- A VNET number to be dialed by the ITG to contact the destination phone. In the case of this call flow this is the VNET number of the destination phone. This allows the call to use the existing VNET translation and routing mechanisms provided by the DAP.

If the VNET number translates to a phone which is reachable through a private ITG connected to the customer's PBX, the directory service will return the following.

- The VNET number of an ITG gateway that is connected to the PBX serving the destination phone. This association between the destination phone the ITG connected to its serving PBX is made by the directory service.

- The VNET number to be dialed by the ITG when it offers the call to the PBX. In most cases this will just be an extension number.

3. PC connects to an ITG

Source Internet PC Telephony Gated a I) IP Telephony Dial « Dialed number 1)Call aci r r g 3) IP Telephony Answer 4)Voice path established 1. A PC uses its telephony software package to send a "connection" message to an ITG.

This IP address is usually returned from the directory service in response to a VNET translation. The specific format and contents of this message is dependent upon the software sending the message or the ITG software to receive the message. This message may contain information identifying the user of the

PC or it may contain information specifying the parameters associated with the requested connection.

2. The ITG responds to the connect message by responding to the message with an acknowledgment that a call has been received. This step of call setup may not be necessary for a PC calling an ITG, but it is shown here in an attempt to maintain a consistent call setup procedure that is independent of whether the PC is connecting to an ITG or to another PC. When connecting to a PC. this step of the procedure allows the calling PC to know that the destination PC is ringing.

3. The ITG accepts the call.

4. A voice path is established between the ITG and the PC.

4. ITG connects to a PC Interact Destination Telephony PC Gateway Offer call I) P Telephony Dial 2) Call Ack Call accept 3) IF Telephony Answer 4)Voice path established

1. An ITG uses its telephony software to send a "connection" message to a PC. The ITG must know the IP address of the PC to which it is connecing. The specific format and contents of this message is dependent upon the ITG software sending the message or the PC software to receive the message. This message may contain information identifying this call as one being offered from an ITG, or it may contain information specifying the requested configuration for the call (i.e. voice only call).

2. The message from step 1 is received by the PC and the receipt of this message is acknowledged by sending a message back to the ITG indicating that the PC is offering the call to the user of the PC 3. The user of the PC answers to call and a message is sent back to the originating PC indicating that the call has been accepted.

4. A voice path is established between the ITG and the PC.

5. VNET PC to PC Call Flow Description The user for PC12 1051 connects the computer to an Internet Protocol (IP) network 1071, turns on the computer and starts an IP telephony software protocol system. The system software transmits a message to a directory service 1031 to register the computer as "on-line" and available to receive calls. This message contains IP address identifying the connection that is being used to connect this computer to the network. This address may be used by other IP telephony software packages to establish a connection to this computer. The address comprises an identification of the computer or virtual private network number that may be used to address this computer 1051. In this VNET scenario, the address is a VNET number assigned to the individual using this PC. VNET refers to a virtual network in which a particular set of telephone numbers is supported as a private network of numbers that can exchange calls. Many corporations currently buy communication time on a trunk that is utilized as a private communication channel for placing and receiving inter-company calls.

The address may also be some identification such as name, employee id, or any other unique ID.

The message may contain additional information regarding the specifics of the system software or the hardware configuration of PCI 11051 utilized for IP telephony. As an example, it is important for a calling PC to know what type of compression algorithms are supported and active in the current communication, or other capabilities of the software or

hardware that might affect the ability of other users to connect or use special feature during a connection.

6. Determining best choice for Internet client selection of an Internet Telephony Gateway server on the Internet: Figure 10B illustrates an internet routing network in accordance with a preferred embodiment. If a client computer 1080 on the Internet needs to connect to an Internet Telephony Gateway 1084, the ideal choice for an Gateway to select can fall into two categories, depending on the needs of the client: If the client 1080 needs to place a telephone call to a regular PSTN phone, and PSTN network usage is determined to be less expensive or higher quality than Internet network usage, it is the preferred choice to select a gateway that allows the client to access the PSTN network from a point "closest" to the point of internet access. This is often referred to as Head-End Hop-Off (HEHO), where the client hops off the internet at the "head end" or "near end" of the internet.

If the client 1080 needs to place a telephone call to a regular PSTN phone, and the PSTN network is determined to be more expensive than Internet network usage, it is the preferred choice to select a gateway that allows the client to access the PSTN from the Internet at a point closest to the destination telephone. This is often referred to as Tail-End Hop-Off (TEHO), where the client hops off the internet at the "tail end" or "far end" of the internet.

a) Head-End Hop-Off Methods (1) Client Ping Method This method selects the best choice for a head-end hop-off internet telephony gateway by obtaining a list of candidate internet telephony gateway addresses, and pinging each to determine the best choice in terms of latency and number of router hops. The process is as follows: Client Computer 1080 queries a directory service 1082 to obtain a list of candidate internet telephony gateways.

The directory service 1082 looks in a database of gateways and selects a list of gateways to offer the client as candidates. Criteria for selecting gateways as candidates can include last gateway selected.

matching 1, 2, or 3 octets in the IPv4 address.

last client access point, if known.

selection of at least one gateway from all major gateway sites, if practical.

The directory service 1082 returns a list of"n" candidate IP addresses to the client 1080 in a TCP/IP message.

The client 1080 simultaneously uses the IP ping to send an echo-type message to each candidate Internet Telephony Gateway, 1084, 1081, 1086. The "-r" option will be used with the ping command to obtain a trace route.

Based upon the ping results for each Internet Telephony Gateway, the client 1080 will rank order the ping results as follows: If any Internet Telephony Gateways are accessible to the client 1080 without traveling through any intervening router as revealed by the ping trace route, they are ranked first.

The remaining Internet Telephony Gateways are ranked in order of lowest latency of round-trip ping results.

Using the Client Ping Method with the Sample Network Topology above, the Client Computer 1080 queries the Directory Service 1082 for a list of Internet Telephony Gateways to ping. The Directory Service 1082 returns the list: 166.37.61.117 166.25.27.101 166.37.27.205 The Client Computer 1080 issues the following three commands simultaneously: ping 166.37.61.117-ri ping 166.25.27.101 -r 1 ping 166.37.27.205 -r 1 The results of the ping commands are as follows: Pinging 166.37.61.117 with 32 bytes of data:

Reply from 166.37.61.117: bytes=32 time=3ms TTL=30 Route: 166.37.61.101 Reply from 166.37.61.117: bytes=32 time=2ms TTL=30 Route: 166.37.61.101 Reply from 166.37.61.117: bytes=32 time=2ms TTL=31 Route: 166.37.61.101 Reply from 166.37.61.117: bytes=32 time=2ms TTL=30 Route: 166.37.61.101 Pinging 166.25.27.101 with 32 bytes of data: Reply from 166.25.27.101: bytes=32 time=14ms TTL=30 Route: 166.37.61.101 Reply from 166.25.27.101: bytes=32 time=2ms TTL=30 Route: 166.37.61.101 Reply from 166.25.27.101: bytes=32 time=3ms TTL=31 Route: 166.37.61.101 Reply from 166.25.27.101: bytes=32 time=4ms TTL=30 Route: 166.37.61.101 Pinging 166.37.27.205 with 32 bytes of data: Reply from 166.37.27.205: byies=32 time=lms TTL=126 Route: 166.37.27.205 Reply from 166.37.27.205: bytes=32 time=lms TTL=126 Route: 166.37. 27.205 Reply from 166.37. 27.205: bytes=32 time=lms TTL=126 Route: 166.37. 27.205 Reply from 166.37. 27.205: bytes=32 time=lms TTL=126

Route: 166.37. 27.205 Since the route taken to 166.37.27.205 went through no routers (route and ping addresses are the same), this address is ranked first. The remaining Internet Telephony Gateway Addresses are ranked by order of averaged latency. The resulting preferential ranking of Internet Telephony Gateway addresses is 166.37.27.205 166.37.61.117 166.25.27.101 The first choice gateway is the gateway most likely to give high quality of service, since it is located on the same local area network. This gateway will be the first the client will attempt to use.

(2) Access Device Location Method The method for identifying the most appropriate choice for an Internet Telephony Gateway utilizes a combination of the Client Ping Method detailed above, and the knowledge of the location from which the Client Computer 1080 accessed the Internet. This method may work well for clients accessing the Internet via a dial-up access device.

A client computer 1080 dials the Internet Access Device. The Access Device answers the call and plays modem tone. Then, the client computer and the access device establishes a PPP session. The user on the Client Computer is authenticated (username/password prompt, validated by an authentication server). Once the user passes authentication, the Access Device can automatically update the User Profile in the Directory Service for the user who was authenticated, depositing the following information "User Name" "Account Code" "online timestamp" "Access Device Site Code" Later, when the Client Computer requires access through an Internet Telephony Gateway, it queries the Directory Service 1082 to determine the best choice of Internet Telephony

Gateway. If an Access Device Site Code is found in the User's Profile on the Directory Service, the

Directory Service 1082 selects the Internet Telephony Gateway 1084, 1081 and 1086 at the same site code, and returns the IP address to the Client Computer 1080. If an Internet Telephony Gateway 1084, 1081 and 1086 is unavailable at the same site as the Access Device Site Code, then the next best choice is selected according to a network topology map kept on the directory server.

If no Access Device Site Code is found on the directory server 1082, then the client 1080 has accessed the network through a device which cannot update the directory server 1082. If this is the case, the Client Ping Method described above is used to locate the best alternative internet telephony gateway 1084.

(3) User Profile Method Another method for selection of an Internet Telephony Gateway 1084, 1081 and 1086 is to embed the information needed to select a gateway in the user profile as stored on a directory server. To use this method, the user must execute an internet telephony software package on the client computer. The first time the package is executed, registration information is gathered from the user, including name, email address, IP Address (for fixed location computers), site code, account code, usual internet access point, and other relevant information. Once this information is entered by the user, the software package deposits the information on a directory server, within the user's profile.

Whenever the Internet Telephony software package is started by the user, the IP address of the user is automatically updated at the directory service. This is known as automated presence notification. Later, when the user needs an Internet Telephony Gateway service, the user queries the directory service for an Internet Telephony Gateway to use. The directory service knows the IP address of the user and the user's usual site and access point into the network. The directory service can use this information, plus the network map of all Internet Telephony Gateways 1084, 1081 and 1086, to select the best Internet Telephony Gateway for the client computer to use.

(4) Gateway Ping Method

The last method selects the best choice for a head-end hop-off internet telephony gateway by obtaining a list of candidate internet telephony gateway addresses, and pinging each to determine the best choice in terms of latency and number of router hops. The process is as follows: Client Computer queries a directory service to obtain a best-choice internet telephony gateway.

The directory service looks in a database of gateways and selects a list of candidate gateways. Criteria for selecting gateways as candidates can include last gateway selected.

matching 1 2, or 3 octets in the IPv4 address

last client access point, if known.

selection of at least one gateway from all major gateway sites, if practical.

The directory sends a message to each candidate gateway, instructing each candidate gateway to ping the client computer's IP Address.

Each candidate gateway simultaneously uses the IP ping to send an echo-type message to the client computer. The "-r" option will be used with the ping command to obtain a trace route. The ping results are returned from each candidate gateway to the directory service.

Based upon the ping results for each Internet Telephony Gateway, the directory service will rank order the ping results as follows: If any Internet Telephony Gateways are accessible to the client without traveling through any intervening router as revealed by the ping trace route, they are ranked first.

The remaining Internet Telephony Gateways are ranked in order of lowest averaged latency of round-trip ping results.

The Client Ping Method and Gateway Ping Method may use the traceroute program as an alternative to the ping program in determining best choice for a head-end hop-off gateway.

b) Tail-End Hop-Off Methods Tail-End Hop-Off entails selecting a gateway as an egress point from the internet where the egress point is closest to the terminating PSTN location as possible. This is usually desired to avoid higher PSTN calling rates. The internet can be used to bring the packetized voice to the local calling area of the destination telephone number, where lower local rates can be paid to carry the call on the PSTN.

(1) Gateway Registration One method for Tail-End Hop-Off service is to have Internet Telephony Gateways 1084, 1081 and 1086 register with a directory service. Each Internet Telephony Gateway will have a profile in the directory service which lists the calling areas it serves. These can be listed in terms of Country Code, Area Code, Exchange, City Code, Line Code, Wireless Cell, LATA, or any other method

which can be used to subset a numbering plan. The gateway, upon startup, sends a TCP/IP registration message to the Directory Service 1082 to list the areas it serves.

When a Client Computer wishes to use a TEHO service, it queries the directory service for an Internet Telephony Gateway 1084 serving the desired destination phone number. The directory service 1082 looks for a qualifying Internet Telephony Gateway, and if it finds one, returns the IP address of the gateway to use. Load-balancing algorithms can be used to balance traffic across multiple Internet Telephony Gateways 1084, 1081 and 1086 serving the same destination phone number.

If no Internet Telephony Gateways 1084, 1081 and 1086 specifically serve the calling area of the given destination telephone number, the directory service 1082 returns an error TCP/IP message to the Client Computer 1080. The Client 1080 then has the option of querying the Directory Service for any Internet Telephony Gateway, not just gateways serving a particular destination telephone number.

As a refinement of this Gateway Registration scheme, Gateways can register calling rates provided for all calling areas. For example, if no gateway is available in Seattle, it may be less expensive to call Seattle from the gateway in Los Angeles, than to call Seattle from the gateway in Portland. The rates registered in the directory service can enable the directory service the lowest cost gateway to use for any particular call.

7. Vnet Call Processing Figure 11 is a callflow diagram in accordance with a preferred embodiment. Processing commences at 1101 where the location of the directory service to receive this "on-line"

message will be determined by the data distribution implementation for this customer. In some cases this may be a private database for a company or organization subscribing to a VNET service, in other cases it might be a national or worldwide database for all customers of a service provider (MCI). When the directory service receives this message from PC12 1051, it will update a profile entry associated with the unique ID to indicate that the user is "on-line' and is located at the specified IP address. Then, at 1102, after successful update of the profile associated with the ID, the directory service sends a response (ACK) back to the specified IP address indicating that the message was received and processed. When the computer (PC12) receives this response message it may choose to notify the user via a visual or audible indicator.

At 1103, a user of PCI 1 1052 connects a computer to an IP network, turns on the computer and starts telephony system software. The registration process for this computer follows the same procedures as those for PC12 1051. In this scenario it is assumed that the directory service receiving this message is either physically or logically the same directory service that received the message from PC12 1051.

At 1104, when the directory service 1031 receives a message from PCI 11052, it initiates a similar procedure as it followed for a message for PC12 1051. However, in this case it will update the profile associated with the identifier it received from PC 111052, and it will use the IP address it received from PC1 1052. Because of the updated profile information, when the acknowledgment message is sent out from the directory service, it is sent to the IP address associated with PCI 11052. At this point both computers (PC12 1051 and PCI 11052) are "on-line" and available to receive calls.

At 1105, PC12 1051 uses its telephony system software to connect to computer PC11 11052.

To establish this connection, the user of PC12 1051 dials the VNET number (or other unique ID such as name, employee ID, etc). Depending upon the implementation of the customer's network, and software package, a unique network identifier may have to be placed in this dial string. As an example, in a telephony implementation of a VNET, a subscriber may be required to enter the number 8 prior to dialing the VNET number to signal a PBX that they are using the VNET network to route the call. Once the telephony software package has

identified this call as a VNET type call, it will send a translation request to the directory service. At a minimum, this translation request will contain the following information: - The IP address of the computer (PC12 1051) sending this request, and 1 The VNET number (or other ID) of the computer to be dialed.

At 1106, when the directory service receives this message, it uses the VNET number (or other ID) to determine if the user associated with the VNET number (or other ID) is "on-line" and to identify the IP address of the location where the computer may be contacted. Any additional information that is available about the computer being contacted (PCI 1 1052), such as compression algorithms or special hardware or software capabilities, may also be retrieved by the directory service 1031. The directory service 1031 then returns a message to PCl2 1051 with status information for PCI 11052, such as whether the computer is "on- line," its IP address if it is available and any other available information about capabilities of PC1 11 1052. When PC12 1051 receives the response, it determines whether PC 11052 may be contacted. This determination will be based upon the "on-line" status of PC1 11052, and the additional information about capabilities of PC 1052. If PC12 1051 receives status information indicating that PC11 1052 may not be contacted, the call flow stops here, otherwise it continues.

The following steps 1107 through 1111 are "normal" IP telephony call setup and tear-down steps. At 1107, PC12 1051 transmits a "ring" message to PCl 11052. This message is directed to the IP address received from the directory service 1031 in step 1106. This message can contain information identifying the user of PC12 1051, or it may contain information specifying the parameters associated with the requested connection.

At 1108, the message from step 1107 is received by PC1 1 1052 and the receipt of this message is acknowledged by sending a message back to PC12 1051 indicating that the user of PC 11052 is being notified of an incoming call. This notification may be visible or audible depending upon the software package and its configurations on PC 111052.

At 1109, if the user of PC11 1052 accepts the call, a message is sent back to PC12 1051 confirming "answer" for the call. If the user of PC 1 1052 does not answer the call or chooses to reject the call, a message will be sent back to PC 12 1051 indicative of the error condition. If the call was not answered, the call flow stops here, otherwise it continues.

At 1110, the users of PC12 1051 and PC11 1052 can communicate using their telephony software. Communication progresses until at 1111 a user of either PC may break the connection by sending a disconnect message to the other call participant. The format and contents of this message is dependent upon the telephony software packages being used by PC12 1051 and PCl 11052. In this scenario, PCl 11052 sends a disconnect message to PC12 1051, and the telephony software systems on both computers discontinue transmission of voice.

Figure 12 illustrates a VNET Personal Computer (PC) to out-of-network PC Information call flow in accordance with a preferred embodiment. In this flow, the Internet telephony gateway is an out-of-network element. This means that the Internet Telephony Gateway cannot use SS7 signaling to communicate with the switch, it must simply outpulse the VNET number to be dialed. An alternate embodiment facilitates directory services to do a translation of the VNET number directly to a Switch/Trunk and outpulse the appropriate digits. Such processing simplifies translation in the switching network but would require a more sophisticated signaling interface between the internet gateway and the switch. This type on "in-network" internet gateway scenano will be covered in another call flow.

This scenario assumes that there is no integration between the internet and a customer premises Public Branch Exchange (PBX). If there were integration, it might be possible for the PC to go through the Internet (or intranet) to connect to an ITG on the customers PBX, avoiding the useof the PSTN. Figure 12 is a callflow diagram in accordance with a preferred embodiment. Processing commences at 1201 where the location of the directory service to receive this "on-line" message will be determined by the data distribution implementation for this customer. In some cases this may be a private database for a company or organization subscribing to a VNET service, in other cases it might be a national or worldwide database for all customers of a service provider (MCI).

When the directory service receives this message from PCl2 1051, it will update a profile entry associated with the unique ID to indicate that the user is "on-line" and is located at the specified IP address. Then, at 1202, after successful update of the profile associated with the ID, the directory service sends a response (ACK) back to the specified IP address indicating that the message was received and processed. When the computer (PC 12) receives this response message it may choose to notify the user via a visual or audible indicator.

At 1203, a VNET translation request is then sent to the directory services to determine the translation for the dial path to the out of network internet gateway phone. A response including the IP address and the DNIS is retumed at 1204. The response completely resolves the phone addressing information for routing the call. Then, at 1205, an IP telephony dial utilizing the DNIS information occurs. DNIS refers to Dialed Number Information Services which is definitive information about a call for use in routing the call. At 1206 an ACK is returned from the IP telephony, and at 1207 an IP telephony answer occurs and a call path is established at 1208.

1209a shows the VNET PC going offhook and sending a dial tone 1209b, and outpulsing digits at 1210. Then, at 1211, the routing translation ofthe DNIS information is used by the routing database to determine how to route the call to the destination telephone. A translation response is received at 1212 and a switch to switch outpulse occurs at 1213. Then, at 1215 a ring is transmitted to the destination phone, and a ringback to the PC occurs. The call is transmitted out of the network via the internet gateway connection and answered at 1216.

Conversation ensues at 1217, until one of the parties hangs up at 1218.

Figure 13 illustrates a VNET Personal Computer (PC) to out-of-network Phone Information call flow in accordance with a preferred embodiment. In this call flow, the use of the PSTN is avoided by routing the call from the PC to the Internet/Intranet to an internet gateway directly connected to a PBX.

Figure 14 illustrates a VNET Personal Computer (PC) to in-network Phone Information call flow in accordance with a preferred embodiment. In this call flow, the internet telephony gateway is an in-network element. This requires that the internet gateway can behave as if it were a switch and utilize SS7 signaling to hand the call off to a switch. This allows the directory service to return the switch/trunk and outpulse digits on the first VNET lookup.

This step avoids an additional lookup by the switch. In this case the directory service must have access to VNET routing information.

a) PC to PC Figure 15 illustrates a personal computer to personal computer internet telephony call in accordance with a preferred embodiment. In step 1501, a net phone user connects through the internet via an IP connection to the step 1502 MCI directory service where a look up is performed to determine how to route the call. In step 1503, the call is terminated in the Intelligent System Platform (ISP) to determine where to send the call. IP Router is the gateway that goes into the MCI ISP to determine via the Intelligent Services Network (ISN) feature engine how to get the call through the network. In step 1504, the call is connected through the Internet to the Net Phone user. In alternative scenario step 1504 the person at the phone is unavailable, so the calling party desired to speak with an MCI operator and the IP Router goes through the Net-Switch (interface to the voice wolld.) In step 1505, the netswitch queries the call processing engine to do DSP Engine functions. In step 1506, the call is routed through the WAN Hub to a MCI switch to an MCI Operator or voicemail in step 1507. This preferred embodiment utilizes the existing infrastructure to assist the call.

b) PC TO PHONE Figure 16, illustrates a phone call that is routed from a PC through the Internet to a phone. In step 1602, the MCI Directory is queried to obtain ISN information for routing the call. Then the call is redirected in step 1603 to the ISP Gateway and routed utilizing the IP router to the call processing engine in steps 1604 and 1605. Then, in step 1606, the call is routed to the WAN and finally to the RBOC where Mainframe billing is recorded for the call.

c) Phone to PC Figure 17 illustrates a phone to PC call in accordance with a preferred embodiment. In step 1701, a phone is routed into a special net switch where in step 1702, a call processing engine determines the DTMF tones utilizing a series of digital signal processors. Then, at step 1703,

the system looks up directory information and connects the call. If the caller is not there, or

busy, then at step 1704, the call is routed via an IP router over the switch utilizing the call processing engine in step 1705.

d) Phone to Phone Figure 18 illustrates a phone to phone call over the internet in accordance with a preferred embodiment. A call comes into the switch at step 1801, and is processed by the call logic program running in the call processing engine in step 1802. In step 1803, a lookup is performed in the directory information database to determine routing of the call as described above. The routing includes storing a billing record in the mainframe billing application 1808. All of the ISN features are available to the call even thought the call is routed through the internet. An IP router is used at each end of the internet to facilitate routing of the call through the internet 1804 and into the network switch. From the network switch the call is routed to a call processing engine through a WAN hub 1806 through the RBOC 1807 to the target telephone. Various DSP Engines 1803 are utilized to perform digital transcoding, DTMF detection, voice recognition, call progress, VRU functions and Modem functions.

XI. TELECOMMUNICATION NETWORK MANAGEMENT A preferred embodiment utilizes a network management system for a telecommunication network for analyzing, correlating, and presenting network events. Modern telecommunications networks utilize data signaling networks, which are distinct from the call-bearing networks, to carry the signaling data that are required for call setup, processing, and clearing. These signaling networks use an industry-standard architecture and protocol, collectively referred to as Common Channel Signaling System #7, or Signaling System #7 (SS7) for short. SS7 is a significant advancement over the previous signaling method, in which call signaling data were transmitted over the same circuits as the call. SS7 provides a distinct and dedicated network of circuits for transmitting call signaling data. Utilizing SS7 decreases the call setup time (perceived by the caller as post-dial delay) and increases capacity on the call-bearing network. A detailed description of SS7 signaling is provided in Signaling System #7, Travis Russell, Mcgraw Hill (1995).

The standards for SS7 networks are established by ANSI for domestic (U.S.) networks, by ITU for international connections, and are referred to as ANSI SS7 and ITU C7, respectively.

A typical SS7 network is illustrated in Figure 1B. A call-bearing telecommunications network makes use of matrix switches 102a/102b for switching customer traffic. These switches 102a/102b are conventional, such as a DMS-250 manufactured by Northern Telecom or a DEX-600 manufactured by Digital Switch Corporation. These switches 102a/102b are interconnected with voice-grade and data-grade call-bearing trunks. This interconnectivity, which is not illustrated in Figure 1B, may take on a large variety of configurations.

Switches in telecommunications networks perform multiple functions. In addition to switching circuits for voice calls, switches must relay signaling messages to other switches as part of call control. These signaling messages are delivered through a network of computers, each of which is called a Signaling Point (SP) 102a/102b. There are three kinds of SPs in an SS7 network: - Service Switching Point (SSP) - Signal Transfer Point (STP) - Service Control Point (SCP) The SSPs are the switch interface to the SS7 signaling network.

Signal Transfer Points (STPs) 104a...104f (collectively referred to as 104) are packet- switching communications devices used to switch and route SS7 signals. They are deployed in mated pairs, known as clusters, for redundancy and restoration. For example, in Fig. 1B, STP 104a is mated with STP 104b in Regional Cluster 1, STP 104c is mated with STP 104d in Regional Cluster 2, and STP 104e is mated with STP 104f in Regional Cluster 3. A typical SS7 network contains a plurality of STP clusters 104; three are shown in Fig. 1 for illustrative purposes. Each STP cluster 104 serves a particular geographic region of SSPs 102. A plurality of SSPs 102 have primary SS7 links to each of two STPs 104 in a cluster. This serves as a primary homing arrangement. Only two SSPs 102 are shown homing to Regional Cluster 2 in Fig. 1B for illustrative purposes; in reality, several SSPs 102 will home on a particular STP cluster 104. SSPs 102 will also generally have a secondary SS7 link to one or both STPs 104 in another cluster. This serves as a secondary homing arrangement.

The SS7 links that connect the various elements are identified as follows:

A links connect an SSP to each of its primary STPs (primarily homing).

B links connect an STP in one cluster to an STP in another cluster.

C links connect one STP to the other STP in the same cluster.

D links connect STPs between different carrier networks (not illustrated).

E links connect an SSP to an STP that is not in its cluster (secondary homing).

F links connect two SSPs to each other.

To interface two different carriers' networks, such as a Local Exchange Carrier (LEC) network with an Interchange Carrier (IXC) network, STP clusters 104 from each carriers' network may be connected by D links or A links. SS7 provides standardized protocol for such an interface so that the signaling for a call that is being passed between an LEC and an IXC may also be transmitted.

When a switch receives and routes a customer call, the signaling for that call is received (or generated) by the attached SSP 102. While intermachine trunks that connect the switches carry the customer's call, the signaling for that call is sent to an STP 104. The STP 104 routes the signal to either the SSP 102 for the call-terminating switch, or to another STP 104 that will then route the signal to the SSP 102 for the call-terminating switch. Another element of an SS7 network are Protocol Monitoring Units (PMU) 106, shown in Figure 2. PMUs 106 are deployed at switch sites and provide an independent monitoring tool for SS7 networks.

These devices, such as those manufactured by INET Inc. of Richardson, TX., monitor the A, E, and F links of the SS7 network, as shown in Figure 2. They generate fault and performance information for SS7 links.

As with any telecommunications network, an SS7 network is vulnerable to fiber cuts, other transmission outages, and device failures. Since an SS7 network carries all signaling required to deliver customer traffic, it is vital that any problems are detected and corrected quickly.

Therefore, there is an essential need for a system that can monitor SS7 networks, analyze fault and performance information, and manage corrective actions.

Prior art SS7 network management systems, while performing these basic functions, have several shortcomings. Many require manual configuration of network topology, which is vulnerable to human error and delay topology updates. Configuration of these systems

usually requires that the system be down for a period of time. Many systems available in the industry are intended for a particular vendor's PMU 106, and actually obtain topology data from their PMUs 106, thereby neglecting network elements not connected to a PMU 106 and other vendors' equipment.

Because prior art systems only operate with data received from proprietary PMUs 106, they do not provide correlation between PMU events and events generated from other types of SS7 network elements. They also provide inflexible and proprietary analysis rules for event correlation.

A system and method for providing enhanced SS7 network management functions are provided by a distributed client/server platform that can receive and process events that are generated by various SS7 network elements. Each network event is parsed and standardized to allow for the processing of events generated by any type of element. Events can also be received by network topology databases, transmission network management systems network maintenance schedules, and system users. Referring to Figure 3, the systems architecture of the preferred embodiment of the present invention, referred to as an SS7 Network Management System (SNMS), is illustrated. SNMS consists of four logical servers 302/304/306/308 and a plurality of client workstations 312a/312b/312c connected via a Network Management Wide Area Network (WAN) 310. The four logical SNMS servers 302/304/306/308 may all reside on a single or a plurality of physical units. In the preferred embodiment, each logical server resides on a distinct physical unit, for the purpose of enhancing performance. These physical units may be of any conventional type, such as IBM RS6000 units running with AIX operating system.

The client workstations 312 may be any conventional PC running with Microsoft Windows or IBM OS/2 operating systems, a dumb terminal, or a VAX VMS workstation. In actuality, client workstations may be any PC or terminal that has an Internet Protocol (IP) address, is running with X-Windows software, and is connected to the WAN 310. No SNMS-specific software runs on the client workstations 312.

SNMS receives events from various SS7 network elements and other network management systems (NMS) 338. It also receives network topology, configuration, and maintenance data

from various external systems, as will be described. The various network elements that generate events include Network Controllers 314, International and Domestic SPs 316/102, STPs 104, and PMUs 106. Network Controllers 314 are devices that switch circuits based on external commands. They utilize SS7 signaling in the same manner as an SSP 102, but are not linked to any STPs 104. International SPs 316 support switches that serve as a gateway between a domestic and international telecommunications network. The STPs 104 may be domestic or international.

The PMUs 106 scan all the SS7 packets that pass across the SS7 circuits, analyze for fault conditions, and generate network events that are then passed onto SNMS. The PMUs 106 also generate periodic statistics on the performance of the SS7 circuits that are monitored.

All SPs 102/316, STPs 104, PMU 106, and SS7 Network Controllers 314 transmit network events to SNMS via communications networks. This eliminates the need for SNMS to maintain a session with each of the devices. In one typical embodiment, as illustrated in Fig.

3, an Asynchronous Data Communications Network 320 is used to transport events from Network Controllers 314 and International SPs 316. An IBM mainframe Front End Processor (FEP) 324, such as IBM's 3708, is used to convert the asynchronous protocol to SNA so it can be received by a IBM mainframe-based Switched Host Interface Facilities Transport (SWIFT) system 326. SWIFT 326 is a communications interface and data distribution application that maintains a logical communications session with each of the network elements.

In this same embodiment, an X.25 Operational Systems Support (OSS) Network 328 is used to transport events from STPs 104, SPs 102, and PMUs 106. These events are received by a Local Support Element (LSE) system 330. The LSE 330, which may be a VAX/VMS system, is essentially a Packet Assembler/ Disassembler (PAD) and protocol converter used to convert event data from the X.25 OSS Network 328 to the SNMS servers 302/304. It also serves the same function as SWIFT 326 in maintaining communication sessions with each network element, thus eliminating the need for SNMS to do so. The need for both SWIFT 326 and LSE 330 illustrates one embodiment of a typical telecommunications network in which different types of elements are in place requiring different transport mechanisms. SNMS supports all these types of elements.

All network events are input to the SNMS Alarming Server 302 for analysis and correlation.

Some events are also input to the SNMS Reporting Server 304 to be stored for historical purposes. A Control system 332, which may be a VAX/VMS system is used to collect topology and configuration data from each of the network elements via the X.25 OSS Network 328. Some elements, such as STPs 104 and SPs 102, may send this data directly over the X.25 OSS Network 328. Elements such as the International SSP 316, which only communicates in asynchronous mode, use a Packet Assembler/ Disassembler (PAD) 318 to connect to the X.25 OSS Network 328. The Control system 332 then feeds this topology and configuration data to the SNMS Topology Server 306.

Network topology information is used by SNMS to perform alarm correlation and to provide graphical displays. Most topology information is received from Network Topology Databases 334, which are created and maintained by order entry systems and network engineering systems in the preferred embodiment. Topology data is input to the SNMS Topology Server 306 from both the Network Topology Databases 334 and the Control System 332. An ability to enter manual overrides through use of a PC 336 is also provided to the SNMS Topology Server 306.

The SNMS Alarming Server 302 also receives events, in particular DS-3 transmission alarms, from other network management systems (NMS) 338. Using topology data, SNMS will correlate these events with events received from SS7 network elements. The SNMS Alarming Server 302 also receives network maintenance schedule information from a Network Maintenance Schedule system 340. SNMS uses this information to account for planned network outages due to maintenance, thus eliminating the need to respond to maintenance- generated alarms. SNMS also uses this information to proactively warn maintenance personnel of a network outage that may impact a scheduled maintenance activity.

The SNMS Alarming Server 302 has an interface with a Trouble Management System 342.

This allows SNMS users at the client workstations 312 to submit trouble tickets for SNMS- generated alarms. This interface, as opposed to using an SNMS-internal trouble management system, can be configured to utilize many different types of trouble management systems. In the preferred embodiment, the SNMS Graphics Server 308 supports all client workstations

312 at a single site, and are therefore a plurality of servers. The geographical distribution of SNMS Graphics Servers 308 eliminates the need to transmit volumes of data that support graphical presentation to each workstation site from a central location. Only data from the Alarming Server 302, Reporting Server 304, and Topology Server 306 are transmitted to workstation sites, thereby saving network transmission bandwidth and improving SNMS performance. In alternative embodiments, the Graphics Servers 308 may be centrally located.

Referring now to Figure 4, a high-level process flowchart illustrates the logical system components of SNMS. At the heart of the process is Process Events 402. This component serves as a traffic cop for SNMS processes. Process Events 402, which runs primarily on the SNMS Alarming Server 302, is responsible for receiving events from other SNMS components, processing these events, storing events, and feeding processed event data to the Reporting and Display components. The Process Events process 402 is shown in greater detail in Figure 5.

The Receive Network Events component 404, which runs primarily on the Alarming Server 302, receives network events from the various SS7 network elements (STPs 104, SPs 102, PMUs 106, etc.) via systems such as SWIFT 326 and LSE 330. This component parses the events and sends them to Process Events 402 for analysis. The Receive Network Events process 404 is shown in greater detail in Figure 6.

The Process Topology component 406, which runs primarily on the Topology Server 306, receives network topology and configuration data from the Network Topology Databases 334, from the SS7 network elements via the Control System 332, and from Manual Overrides 336. This data is used to correlate network events and to perform impact assessments on such events. It is also used to provide graphical presentation of events. Process Topology 406 parses these topology and configuration data, stores them, and sends them to Process Events 402 for analysis. The Process Topology process 406 is shown in greater detail in Figure 7.

The Define Algorithms component 408, which runs primarily on the Alarming Server 302, defines the specific parsing and analysis rules to be used by SNMS. These rules are then loaded into Process Events 402 for use in parsing and analysis. The algorithms are kept in a software module, and are defined by programmed code. A programmer simply programs the pre-defined algorithm into this software module, which is then used by Process Events 402.

These algorithms are procedural in nature and are based on network topology. They consist of both simple rules that are written in a proprietary language and can be changed dynamically by an SNMS user, and of more complex rules which are programmed within SNMS software code.

The Receive NMS Data component 410, which runs primarily on the Alarming Server 302, receives events from other network management systems (NMS) 336. Such events include DS-3 transmission alarms. It also receives network maintenance events from a Network Maintenance Schedule system 340. It then parses these events and sends them to Process Events 402 for analysis. The Display Alarms component 412, which runs primarily on the Graphics Server 308 and the Alarming Server 302, includes the Graphical User Interface (GUI) and associated software which supports topology and alarm presentation, using data supplied by Process Events 402. It also supports user interactions, such as alarm clears, acknowledgments, and trouble ticket submissions. It inputs these interactions to Process Events 402 for storing and required data updates. The Display Alarms process 412 is shown in greater detail in Figure 8.

The Report On Data component 414, which runs primarily on the Reporting Server 304, supports the topology and alarm reporting functions, using data supplied by Process Events 402. The Report On Data process 414 is shown in greater detail in Figure 9.

Referring now to Figure 5, the detailed process of the Process Events component 402 is illustrated. This is the main process of SNMS. It receives generalized events from other SNMS components, parses each event to extract relevant data, and identifies the type of event. If it is an SS7-related event, Process Events 402 applies a selected algorithm, such as create alarm or correlate to existing alarm.

The first three steps 502-506 are an initialization process that is run at the start of each SNMS session. They establish a state from which the system may work. Steps 510-542 are then run as a continuous loop.

In step 502, current topology data is read from a topology data store on the Topology Server 306. This topology data store is created in the Process Topology process 406 and input to Process Events 402, as reflected in Figure 4. The topology data that is read has been parsed in Process Topology 406, so it is read in step 502 by Process Events 402 as a standardized event ready for processing.

In step 504, the algorithms which are created in the Define Algorithms component 408 are read in. These algorithms determine what actions SNMS will take on each alarm. SNMS has a map of which algorithms to invoke for which type of alarm.

In step 506, alarms records from the Fault Management (FM) reporting database, which is created in the Report on Data process 414, are read in. All previous alarms are discarded.

Any alarm that is active against a node or circuit that does not exist in the topology (read in step 502) is discarded. Also, any alarm that does not map to any existing algorithm (read in step 504) is discarded. The alarms are read from the FM reporting database only within initialization. To enhance performance of the system, future alarm records are retrieved from a database internal to the Process Events 402 component. Step 506 concludes the initialization process; once current topology, algorithms, and alarms are read, SNMS may begin the continuous process of reading, analyzing, processing, and storing events.

This process begins with step 510, in which the next event in queue is received and identified.

The queue is a First In/First Out (FIFO) queue that feeds the Process Events component 402 with network events, topology events, and NMS events. To reiterate, the topology data that are read in step 502 and the alarm data that are read in step 504 are initialization data read in at startup to create a system state. In step 510, ongoing events are read in continuously from process components 404, 406, and 410. These events have already been parsed, and are received as standardized SNMS events. SNMS then identifies the type of event that is being received. If the event is found to be older than a certain threshold, for example one hour, the event is discarded.

In steps 512, 520, 524, and 534 SNMS determines what to do with the event based on the event type identification made in step 510.

In step 512, if the event is determined to be topology data, SNMS updates the GUI displays to reflect the new topology in step 514. Then in step 516, SNMS performs a reconciliation with active alarms to discard any alarm not mapping to the new topology. In step 518, the new topology data is recorded in a topology data store, which is kept in the SNMS Topology Server 306.

In step 520, if the event is determined to be NMS data, such as DS-3 alarms 338, it is stored in the FM reporting database on the SNMS Reporting Server 304 for future reference by SNMS rules.

In step 524, if the event is determined to be a defined SS7 network event, then in step 526 one or more algorithms will be invoked for the event. Such algorithms may make use of data retrieved from Network Management Systems 338, Network Maintenance Schedules 340, and Network Topology 334.

For example, when each circuit level algorithm generates an alarm, it performs a check against the Network Maintenance Schedule 340 and NMS 338 records. Each alarm record is tagged if the specified circuit is within a maintenance window (Network Maintenance Schedule 340) or is transported on a DS-3 that has a transmission alarm (NMS 338). While SS7 circuits run at a DS-0 level, the Network Topology Databases 334 provide a DS-3 to DS- 0 conversion table. Any DS-0 circuit within the DS-3 is tagged as potentially contained within the transmission fault. Clear records from NMS 338 will cause active SNMS circuit level alarms to be evaluated so that relevant NMS 338 associations can be removed. SNMS clear events will clear the actual SNMS alarm. GUI filters allow users to mask out alarms that fit into a maintenance window or contained within a transmission fault since these alarms do not require SNMS operator actions.

In step 528, active alarms are reconciled with new alarm generations and clears resulting from step 526. In step 530, the GUI displays are updated. In step 532, the new alarm data is stored in the FM reporting database.

In step 534, the event may be determined to be a timer. SNMS algorithms sometimes need to delay further processing of specific conditions for a defined period of time, such as for persistence and rate algorithms. A delay timer is set for this condition and processing of new SNMS events continues. When the time elapses, SNMS treats the time as an event and performs the appropriate algorithm.

For example, an SS7 link may shut down momentarily with the possibility of functioning again within a few seconds, or it may be down for a much greater period of time due to a serious outage that requires action. SNMS, when it receives this event, will assign a timer of perhaps one minute to the event. If the event clears within one minute, SNMS takes no action on it. However, if after the one minute timer has elapsed the event is unchanged (SS7 link is still down), SNMS will proceed to take action.

In step 536, the appropriate algorithm is invoked to take such action. In step 538, active alarms are reconciled with those that were generated or cleared in step 536. In step 540, the GUI displays are updated. In step 542, the new alarm data is stored in the FM reporting database. As stated previously, SNMS operates in a continuous manner with respect to receiving and processing events. After the data stores in steps 518, 522, 532, and 542, the process returns to step 510.

Referring now to Figure 6, the detailed process of the Receive Network Events component 404 is illustrated. This component collects events from the SS7 network elements via data transport mechanisms, such as the Async Data Network 320, SWIFT 326, X.25 OSS network 328, and the LSE 330. These events are received by the SNMS Alarming Server 302 in a First In/First Out (FIFO) queue. In steps 602 and 604, events from the SS7 network elements are collected by mainframe applications external to SNMS, such as SWIFT 326 and LSE 330, and the protocol of the event data is converted from the network element-specific protocol to SNA or TCP/IP. In one embodiment, SNMS may also have software running on the mainframe that converts the protocol to that recognizable by the SNMS Alarming Server 302.

The event data is then transmitted via SNA or TCP/IP to the SNMS Alarming Server 302.

SNMS maintains a Signaling Event List 608 of all SS7 event types that is to be processed. In step 606, SNMS checks the Signaling Event List 608 and if the current event is found in the list, SNMS traps the event for processing. If the event is not found in the list, SNMS discards it.

In step 610 the event is parsed according to defined parsing rules 614. The parsing rules 614 specify which fields are to be extracted from which types of events, and are programmed into the SNMS code. The parsing of events in step 610 extracts only those event data fields needed within the alarm algorithms or displays. Also input to step 610 are scheduled events 612 from the Network Maintenance Schedule 340. Scheduled events 612 are used to identify each network event collected in step 602 that may be a result of scheduled network maintenance. This allows SNMS operators to account for those SS7 network outages that are caused by planned maintenance.

In step 616, the parsed event data is used to create standardized event objects in SNMS resident memory for use by other SNMS processes. Such event objects are read into the main process, Process Events 402, in step 510.

Referring now to Figure 7, the detailed process of the Process Topology component 406 is illustrated. This process component retrieves network topology and configuration data from three types of sources, creates standardized topology data records, and stores this data for use by other SNMS processes. In particular, it feeds active topology data to Process Events 402, running on the Alarming Server 302, in step 502.

In step 702, the SNMS Topology server 306 collects topology data from three different sources. It collects current connectivity and configuration data generated by the SS7 network elements via the Control system 332. It collects topology data that has been entered into order entry and engineering systems and stored in Network Topology Databases 334. It also accepts manual overrides 336 via workstation. The collection of data from the Topology Database 334 and the Control system 332 occurs on a periodic basis, and is performed independently of the SNMS Alarming server 302. Unlike prior art systems that use data retrieved from PMUs 106, SNMS receives topology data from all types of network elements, including those that are not connected to a PMU 106 such as that of Figure 2. SNMS also uses data reflecting the topology of foreign networks, such as those of a Local Exchange Carrier (LEC) or an international carrier. This data is used to perform impact assessments that will allow the SNMS user to determine facts such as which end customers may be impacted by an SS7 link outage. The type of topology data collected and used by SNMS, and for example, for the SS7 linkage of an STP 104 with a Switch/SSP 102, data is received by

network order entry and engineering systems. The data and a brief description of its contents is provided below.

STP Link ID Identifies each SS7 link to the STP Switch Link ID Identifies each SS7 link to the Switch/SP STP Linkset Identifies a trunk grouping of SS7 links to the STP Switch Linkset Identifies a trunk grouping of SS7 links to the Switch/SP MCI/Telco Circuit ID Identifies the SS7 link to external systems. For interfaces between two different networks, each ID (MCI ID and Telco ID) provides an identification of the SS7 link for each network (MCI and a Telco in this example).

Link Type Identifies the type of SS7 link SLC Signal Link Code For the switched voice network supported by SS7, data is received by network order entry and engineering systems and used to perform SS7 event impact assessments: Voice Trunk Groups Voice trunk group supported by each SSP 102 For the SS7 linkage of a domestic STP 104g to an international STP 104h, data is received by network order entry and engineering systems: Circuit ID Identifies the SS7 link to external systems SLC Signal Link Code For the purpose of performing impact assessments, Local Exchange Carrier (LEC) NPA/NXX assignments and End Office to Access Tandem homing arrangements are received by a calling area database that is populated by Bellcore's Local Exchange Routing Guide (LERG).

LATA Local Access Transport Area (conventional)

NPA/NXX Numbering Plan Area/prefix (conventional) End Office LEC customer serving node Access Tandem LEC end office hub Foreign network STP 104 clustering and SSP 102 homing arrangements are received by SS7 network elements via a control system.

Point Code Identifies SS7 node (conventional) Data identifying certain aspects of each network element are received by a Switch Configuration File, which resides in an external system.

Data mapping each network DS-0 onto a DS-3 is received by Network Topology Databases.

This data is used to assign DS-3 alarms received by NMS to DS-0 level circuits.

Data needed to overwrite data acquired through automated processes are provided by manual overrides.

Referring now back to Figure 7 in step 704, the various topology data are parsed to extract the data fields that are needed by SNMS algorithms. The data are then standardized into event records that can be processed by Process Events 402.

In step 706, the standardized data records are validated against other data. For example, circuit topology records are validated against node topology records to ensure that end nodes are identified and defined.

In step 708, the topology data are stored on the Topology server 306 of Figure 3 in a relational database, such as that offered by Sybase.

In step 710, the new topology records are passed from the Topology server 306 to the main SNMS process running on the Alarming server 302 and compared against the active configuration (i.e. configuration that is currently loaded into memory). Active alarm and GUI displays are reconciled to remove alarms that pertain to non-existent topology entries.

In step 712, the topology is stored on the Alarming Server 302 (for use by Process Events 402) in the form of flat files for performance reasons. At this time the flat file mirrors the Topology server 306 database from step 708. This flat file is only accessible by the main process. In step 714, the new topology records are loaded into active SNMS memory and new processes which require topology data now use the new configuration.

Referring now to Figure 8, the detailed process of the Display Alarms component 412 is illustrated. This process component provides the results of SNMS processing to the user (referred to as the "operator"), and accepts operator input as actions to be performed within SNMS. Therefore, the process between Display Alarms 412 and Process Events 402 is two- way. It is important to note that while there is a single Process Events process 402 running for the entire SNMS system, there is a different instance of the Display Alarms process 412 running for each operator that is logged onto SNMS. That is, each operator instigates a separate execution of Display Alarms 412.

When an operator logs on SNMS, the first four steps, 802 - 808, execute as an initialization.

From there, steps 810 - 838 operate as a continuous loop. The initialization provides each operator with a system state from which to work. In step 802, the current topology is read in and displayed via Graphical User Interface (GUI). Each operator has its own GUI process that is initialized and terminated based upon an operator request. Each GUI process manages its displays independently. Any status change is handled by the individual processes.

In step 804, a filter that defines the specific operator view is read in. Each operator can define the view that his/her GUI process will display. Filter parameters include: 1. Traffic Alarms, Facility alarms, or both 2. Acknowledged Alarms, Unacknowledged Alarms, or both 3. Alarms on circuits within maintenance windows, Alarms on circuits that are not within a maintenance window, or both.

4. Alarms on circuits that have associated transmission alarms (DS-3 alarms via outage ids), Alarms on circuits that do not have associated transmission alarms, or both.

5. Alarms with a specified severity.

6. Alarms on nodes/circuits owned by a specified customer id.

7. Alarms on International circuits, Alarms on Domestic circuits, or both.

The operator's GUI displays are updated both upon initialization in step 804 and when filter changes are requested in steps 828 and 830. Each specific operator's instance of the Display Alarms 412 process opens a connection with Process Events 402 so that only alarm records relevant to the specific operator's filter are transmitted. In step 806, the specific operator's process registers itself with Process Events 402 to identify which alarms are to be sent. In step 808, the GUI display is presented to the operator.

The continuous execution of Display Alarms 412 begins in step 810. Each event that is to be retrieved and presented, as defined by the operator filter, is received and identified. In steps 812, 816, 820, 826, and 836 SNMS determines what to do with the event based on the event type identification made in step 810. In steps 812 and 816, if the event is determined to be an alarm update or a topology update, the operator's GUI display is updated to reflect this, in steps 814 and 818, respectively. Then the next event is received, in step 810.

In step 820, if the event is determined to be an operator action, two activities are required.

First, in step 822, the operator's GUI display is updated to reflect the status change. Then, in step 824, a status change update is sent to the main process, Process Events 402, so that the status change may be reflected in SNMS records and other GUI processes (for other operators) can receive and react to the status changes.

In step 826, if the event is determined to be an operator display action, then it is determined if the action is a filter change request or a display request. In step 828, if it is determined to be a filter change request, then in step 830 the GUI process registers with Process Events 402 so that the appropriate alarms records are transmitted. In step 832, if it is determined to be an operator display request, then in step 834 the requested display is presented to the operator.

Display requests may include: 1. node detail and connection 2. circuit connection 3. linkset connection

4. unknown topology alarms (alarms on objects that are not defined in the topology databases) 5. STP pair connections 6. Nodes contained within a LATA 7. Home/Mate connections (for non-adjacent nodes) 8. NPA/NXX lists 9. trunk group lists 10. end office access tandem 11. rules definition help screens (aid the operator in understanding the actual algorithm used in generating the alarm 12. recommended actions (operator defined actions that should be taken when a specific alarm is received) In step 836, if the event is determined to be a termination request, then the specific operator's GUI process is terminated in step 838. Otherwise, the next event is received in step 810.

Within the Display Alarm process, SNMS provides several unique display windows which support fault isolation, impact assessments, and trouble handling. All of the GUI displays which contain node and circuit symbols are active windows within SNMS (i.e. screens are dynamically updated when alarm status of the node or circuit change). All the displays are possible due to the set of MCI topology sources used within SNMS. SNMS has extensive topology processing of SNMS which is used in operator displays.

A. SNMS Circuits Map This window displays topology and alarm status information for a selected linkset. As network events are received, SNMS recognizes the relationships between endpoints and isolates the fault by reducing generated alarms. This display allows the operator to monitor a linkset as seen from both sides of the signaling circuit (from the perspective of the nodes).

B. SNMS Connections Map This window presents a cluster view of MCI's signaling network. All MCI and non-MCI nodes connected to the MCI STPs in the cluster are displayed along with the associated

linksets. A cluster view is important since a single STP failure/isolation is not service impacting, but a cluster failure is since all MCI SPs have connectivity to both MCI STPs in the cluster.

C. SNMS Nonadjacent Node Map This window presents a STP pair view of a selected LEC signaling network. All LEC SPs, STPs, and SCPs (with signaling relationships to the MCI network) connected LEC STP pair are displayed. MCI's area of responsibility does not include the LEC STP to LEC SSP signaling links, so no linksets are displayed here. This display allows the SNMS operator to monitor a LEC signaling network as seen by the MCI nodes.

D. SNMS LA TA Connections Map This window presents a map of all LEC owned nodes that are located within a specified LATA. As well, the MCI STP pair that serves the LATA is also displayed along with the associated linksets (where applicable). This display allows the operator to closely monitor a specific LATA if/when problems surface within the LATA. LATA problems, while outside MCI's domain of control, can introduce problems within the MCI network since signaling messages are shared between the networks. As well, MCI voice traffic which terminates in the specified LATA can be affected by LATA outages.

E. NPA-NXXInformation List This window presents a list of NPX-NXX's served by a specified LEC switch. This display is very valuable during impact assessment periods (i.e. if the specified LEC switch is isolated, which NPA-NXX's are unavailable).

F. End Office Information List This window presents a list of LEC end office nodes which are homed to the specific LEC access tandem. This display is very valuable during impact assessment periods (i.e. if the specified LEC tandem switch is isolated, which end offices are unavailable).

G. Trunk Group Information List This window presents a list of MCI voice trunks, connected to a specified MCI switch, and the LEC end office switches where they terminate. This display is very valuable during impact assessment periods (i.e. what end offices are impacted when a MCI switch is isolated). This display is also available for selected LEC end office switches.

H. Filter Definition Window The SNMS operator can limited the scope of his displays to: type of alarms that should be presented severity of alarms that should be presented acknowledged alarms, unacknowledged alarms, or both alarms on circuits inside a planned outage window, alarms on circuits outside a planned outage window or both alarms that are not the result of a specified transmission network outage alarms on specified customer nodes or alarms on circuits connected to specified customer I. Trouble Ticket Window The SNMS operator can open trouble tickets on signaling alarms. These trouble tickets are opened in MCI's trouble ticketing system. Operators can also display the status of existing trouble tickets.

Referring now to Figure 9, the detailed process of the Report On Data component 414 is illustrated. This process component, which runs on the Reporting server 304, stores SNMS- processed data and provides reports.

Standardized Network Element (NE) Event Records 914 are received with location specific time stamps. In step 902, the time stamps are converted into Greenwich Mean Time (GMT) so that standardized reports can be produced.

In step 904, all data received are stored in individual database tables. Data may also be archived for long-term storage to tape or disk. This data includes SNMS-generated alarms 916, standardized topology records 918, and performance statistics from PMUs 920. It may also include non-processed data, such as DS-3 alarms from NMS 338 and network maintenance schedule data 340.

In step 906, reports are produced. These reports may be custom or form reports. They may also be produced on demand, or per a schedule. These reports may be presented in a number of ways, including but not limited to electronic mail 908, X-terminal displays 910, and printed reports 912.

XII. VIDEO TELEPHONY OVER POTS The next logical step from voice over the POTS is video. Today, computers are capable of making video "calls" to each other when connected to some type of computer network.

However, most people only have access to a computer network by making a call from their modem on the POTS with another modem on a computer connected to a network, so that they can then "call" another computer on the network, which is in turn connected by a modem to another network computer. It is much simpler (and efficient) to call another person directly on the POTS and have the modems communicate with each other, without network overhead.

ITU recommendation H.324 describes terminals for low bitrate (28.8kbps modem) multimedia communication, utilizing V.34 modems operating over the POTS. H.324 terminals may carry real-time voice, data, and video, or any combination, including video telephony. H.324 terminals may be integrated into personal computers or implemented in stand-alone devices such as videotelephones and televisions. Support for each media type (voice, data, video) is optional, but if supported, the ability to use a specified common mode of operation is required, so that all terminals supporting that media type can interwork.

H.324 allows more than one channel of each type to be in use. Other Recommendations in the H.324 series include the H.223 multiplex (combination of voice, data and video), H.245 control, H.263 video codec (digital encoder and decoder), and G.723.1.1 audio codec.

H.324 makes use of the logical channel signaling procedures of ITU Recommendation H.245, in which the content of each logical channel is described when the channel is opened.

Procedures are provided for allowing each caller to utilize only the multimedia capabilities of

their machine. For example a person trying to make a video (and audio) call to someone who only has audio and not video capabilities can still communicate with the audio method (G.723.1.1) H.324 by definition is a point-to-point protocol. To conference with more than one other person an MCU (Multipoint Control Unit) is needed to act as a video-call bridge. H.324 computers may interwork with H.320 computers on the ISDN, as well as with computers on wireless networks.

A. Components of Video Telephony System 1. DSP modem pools with ACD.

A Digital Signal Processor (DSP) modem pool is a modem bank, with each modem having the ability to be programmed for extra functions (like new V. modem protocols, DTMF detection, etc.) A call is routed from the MCI switch to an ACD. The ACD keeps a matrix of which DSP modems are available. The ACD also communicates with the ISNAP which does a group select to determine which group of Agents are responsible for this call and also which of the agents are free to process this call. In an alternative embodiment, DSP resources can be deployed without an ACD, directly connected to a switch. In this embodiment the DSP resources are managed using an NCS-based routing step.

2. Agent An Agent can be a human Video Operator (video capable MTOC), or an Automated program (video ARU). The ACD knows which Agent ports are available and connects an Agent to an Agent Port.

3. Video on Hold Server If the ACD has no Agent ports available, then the caller is connected to the Video On Hold Server, which has the ability to play advertisements and other non-interactive video, until the ACD finds a free Agent port.

4. Video Mail Server Video-mail messages are stored here. Customers can manage their mail and record greetings to be stored on this server.

5. Video Content Engine Video On Demand content resides on the Video Content Engine. Video stored here can be previously recorded video-conferences, training videos, etc.

6. Reservation Engine When people want to schedule a multi-party video-conference, they can specify the participants and time of the conference on this system. Configuration can be done with the help of a

human Video Operator or by some other form entry method.

7. Video Bridge Because H.324 is a point-to-point protocol, a Multi-point Conferencing Unit (MCU) needs to manage each participants call and re-direct the video streams appropriately. MCU conferencing will be available for customers with H.324 and H.320 compliant systems.

B. Scenario A computer or set-top TV has H.324 compliant software, and a modem for use over POTS, most likely to be 28.8kbps (V.34) or higher. One objective is to call another party. If they do not answer or are busy, the originator has the option of leaving video-mail for the destination party. Another objective is to schedule and participate in a conference with more than two participants.

C. Connection Setup Figure 19B illustrates a call connection setup in accordance with a preferred embodiment.

There are three methods for making a video-call to someone. The first method is to simply call them (from 1 and 7 of Figure 19B. If the destination is busy or doesn't answer, then the

caller can make another call to 1 800 VID MAIL and perform the appropriate procedures as follows.

When a user dials "1 800 VID MAIL" at 1, the ACD on the DSP modem pool will connect a switch to a modem 2 and a port to an Agent 3. Then the user logs-in to the system with a special, custom terminal program that utilizes the data stream part of the H.324 bandwidth (using the ITU T. 120 standard), called the V-mail Data Interface (VMDI). From a graphical user interface, icon or other menu, the caller can choose to - browse and search a directory of video-capable MCI customers, - call another H.324 compliant software program, - create a video-mail for Store & Forward for later delivery, - personalize and record their video-mail greeting messages, - view and manage their video-mail, or - view selections from a library of recordings (Video On Demand).

In an alternate embodiment, a user can dial "1 800 324 CALL" to call a number. Then, if the destination number was 1319 375 1772, the modem dial string would be "ATDT 1 800 324 CALL ,,, 1 319 375 1772" (the comma ',' tells the modem to do a short pause while dialing.) When the connection to 1 800 324 CALL is made, a connection is made from the originator, to an MCI switch 1, to an ARU 5a, selected by an ACD 2a, 3a.

The ARU 5a detects DTMF tones entered through a telephone keypad or other device for generating DTMF tones to get the destination number. The originator remains on hold while the ARU 5a makes a separate call to the destination number 5a, 6a and 7. If the destination answers, the originator is connected to the destination, both party's modems can connect, and the ARU 5a is released. If the destination is busy or does not answer, the call is transferred to 1 800 VID MAIL or an Agent through the DSP modem pool 2. If there are no DTMF tones detected, the call is transferred to an Agent through the DSP modem pool 2. The Agent will make an H.324 connection with the caller and ask for their destination number (or provide help.) The architecture for this alternative is similar to how faxes are detected and transmitted in the directlineMCI system as discussed with respect to an alternative embodiment.

D. Calling the Destination When the destination number is known, the Video On Hold Server provides the video input for the H.324 connection 4. A new call is made from the Agent 5, 6 to the destination number 7. One concern that required analysis while working out a detailed embodiment required determining if modems could re-synchronize after a switch operation without going off-line. If the destination number answers and is a modem, a connection MUST be made at the same speed as the originator modem speed. After modem handshaking is performed, the ACD instructs the switch to release the agent 3, 5 and releases the modems 2 and 6 and connects the originator to the destination 1 and 7. The destination PC realizes that the connection is an H.324 call (not a fax or otherwise) and the video-call proceeds.

In an alternate embodiment, if the destination answers and is a modem, a connection is made.

Then, two H.324 calls are using two DSP modems. The Agent can be released from both calls 3 and 5. The incoming data from each call is copied to the other call 2 and 6. This way, an Agent can monitor the video call for Video Store & Forward 9. When one connection drops carrier, the video-call is complete, and the modem carrier for the remaining call is dropped.

E. Recording Video-Mail, Store & Forward Video and Greetings If a destination number does not answer or is busy, the Video Mail Server will play the appropriate Video-Mail greeting for the owner of the destination number 8. The caller then leaves a video-message, which is stored on the Video Mail Server. The recording of video for Store & Forward Video is exactly the same as leaving a video-message, described above.

Parameters such as destination number, forwarding time, and any other audio S&F features currently available are entered through the VMDI or communicated with a human video operator (or automated video ARU.) To record a personalized greeting for playback when someone cannot reach you because you are busy or do not answer, is similar to leaving Video-Mail. The option to do this is done through the VMDI or communicated with a human video operator.

F. Retrieving Video-Mail and Video On Demand Users have the choice of periodically polling their video-mail for new messages, or have the video-mail server call them periodically when they have a new message waiting.

Configuration is done through the VMDI or human video operator. Managing and checking video-mail is also performed through the VMDI or communicated with a human video operator.

Choice of video to view for Video On Demand (VOD) is through the VMDI. These videos can be previously recorded video-conferences, training videos, etc. and are stored on the Video Content Engine 9.

G. Video-conference Scheduling A user can navigate through the VMDI or Internet 10 WWW forms, or communicate with a human video operator to schedule a multi-point conference. This information is stored on the Reservation Engine 11. The other conference participants are notified of the schedule with a video-mail, e-mail message or otherwise. There will be an option to remind all registered conference participants at a particular time (e.g. 1 hour before the conference), through video- mail (or e-mail, voice-mail, paging service or any other available notification method). The MCU (video bridge) can call each participant 12, or H.324 users can dial In to the MCU at the scheduled time 12.

XIII. VIDEO TELEPHONY OVER THE INTERNET Figure 19E illustrates an architecture for transmitting video telephony over the Internet in accordance with a preferred embodiment. Real-time Transmission Protocol (RTP) based video-conferencing refers to the transmission of audio, video and data encapsulated as RTP messages. For a RTP-based video-conferencing session, a end-user station first establishes a dial-up Point-to-Point (PPP) connection with the Internet which is then used to transport the RTP messages. Audio information is compressed as per G.723.1.1 audio codec (coder - decoder) standards. Video is compressed in accordance with ITU H.263 video codec standards and data is transmitted as per ITU-T 120 standards.

RTP is a protocol providing support for applications with real-time properties. While UDP/IP is its initial target networking environment, RTP is transport-independent so that it can be used over IPX or other protocols. RTP does not address the issue of resource reservation or quality of service control; instead, it relies on resource reservation protocols such as RSVP.

The transmission service with which most network users are familiar is point-to-point, or unicast service. This is the standard form of service provided by networking protocols such as HDLC and TCP.

Somewhat less commonly used (on wire-based networks, at any rate) is broadcast service.

Over a large network, broadcasts are unacceptable (because they use network bandwidth everywhere, regardless of whether individual sub-nets are interested in them or not), and so they are usually restricted to LAN-wide use (broadcast services are provided by low-level network protocols such as IP). Even on LANs, broadcasts are often undesirable because they require all machines to perform some processing in order to determine whether or not they are interested in the broadcast data.

A more practical transmission service for data that is intended for a potentially wide audience is multicast. Under the multicast model on a WAN, only hosts that are actively interested in a particular multicast service will have such data routed to them; this restricts bandwidth consumption to the link between the originator and the receiver of multicast data. On LANs, many interface cards provide a facility whereby they will automatically ignore multicast data in which the kernel has not registered an interest; this results in an absence of unnecessary processing overhead on uninterested hosts.

A. Components RSVP Routers with MBONE capability for broadcast of video from the Video Content Engine and the MCI Conference Space network. MCI will have an MBONE network that multicasts locally and transmits many unicasts out the Internet.

RSVP is a network control protocol that will allow Internet applications to obtain special qualities-of-service (QOS's) for their data flows. This will generally (but not necessarily) require reserving resources along the data path(s) either ahead of time or dynamically. RSVP

is a component of the future "integrated services" Internet, which provides both best-effort and real-time qualities of service. An embodiment is presented in the detailed specification that follows.

When an application in a host (end system) requests a specific QOS for its data stream, RSVP is used to deliver the request to each router along the path(s) of the data stream and to maintain router and host state to provide the requested service. Although RSVP was developed for setting up resource reservations, it is readily adaptable to transport other kinds of network control information along data flow paths.

1. Directory and Registry Engine When people are connected to the internet (whether through modem dial-up, direct connection or otherwise), they can register themselves in this directory. The directory is used to determine if a particular person is available for conferencing.

2. Agents An Agent can be a human Video Operator (video capable MTOC), or an Automated program (video ARU). An Internet ACD in accordance with a preferred embodiment is designed so that Agent ports can be managed. The ACD will know which Agent ports are available and connects an Agent to an available Agent Port. If the ACD has no Agent ports available, then the caller is connected to the Video On Hold Server, which has the ability to play advertisements and other non-interactive video, until the ACD finds a free Agent port.

3. Video Mail Server Video-mail messages are stored here. Customers can manage their mail and record greetings to be stored on this server.

4. Video Content Engine Video On Demand content resides on the Video Content Engine. Video stored here may be previously recorded video-conferences, training videos, etc.

5. Conference Reservation Engine When people want to schedule a multi-party video-conference, they can specify the participants and time of the conference on this system. Configuration can be done with the help of a human Video Operator or by some other form entry method.

6. MCI Conference Space This is the virtual reality area that customers can be present in. Every participant is personified as an "avatar". Each avatar has many abilities and features, such as visual identity, video, voice, etc. Avatars interact with each other by handling various objects that represent document sharing, file transferring, etc., and can speak to each other as well as see each other.

7. Virtual Reality Space Engine The Conference Spaces are generated and managed by the Virtual Reality Engine. The virtual reality engine manages object manipulation and any other logical descriptions of the conference spaces.

B. Scenario If a user has a current connection to the Internet. The user will utilize H.263 compliant system software utilizing RTP (as opposed to TCP) over the Internet. If the user also desires to participate in VR MCI conference-space, and create/view video-mail, the user can join a VR session.

C. Connection Setup The simplest way to make a video call to another person on the Internet is to simply make the call without navigating through menus and options as an initial telephone call. However, if the destination is busy or not answering, MCI provides services for depositing messages.

A customer can login to a telnet server (e.g. telnet vmail.mci.com), or use a custom-made client, or the WWW (e.g. http://vmail.mci.com). The services menu is referred to as the V- Mail Data Interface (VMDI), similar to the VMDI available when dialing through POTS as described above.

From a menu, the caller can choose to: - browse and search a directory of video-capable MCI customers - call another H.263 compliant software program, - create a video-mail for Store & Forward for later delivery, - personalize and record their video-mail greeting messages, - view and manage their video-mail, and - view selections from a library of recordings (Video On Demand).

When a user has specified a party to call by indicating the destination's name, IP address or other identification, the Directory is checked. It is possible to determine if a destination will accept a call without actually calling; so, since it can be determined that the destination will accept a call, the originator's video client can be told to connect to the destination. If the callers are using a WWW browser (e.g. Netscape Navigator, Microsoft Internet Explorer, internetMCI Navigator, etc.) to access the VMDI, then a call can be automatically initiated using Java, JavaScript or Helper App. If a call cannot be completed, there will be a choice to leave video-mail.

D. Recording Video-Mail, Store & Forward Video and Greetings If an Agent determines that a destination party is not available (off-line, busy, no answer, etc.), the Video Mail Server plays an appropriate Video-Mail greeting for the owner of the destination number 8. The caller then leaves a video-message, which is stored on the Video Mail Server. The recording of video for Store & Forward (S&F) Video is exactly the same as leaving a video-message, described above. Parameters such as destination number, forwarding time, and any other audio S&F features currently available are entered through the VMDI or communicated with a human video operator (or automated video ARU.)

Customers may record their own personalized greetings to greet callers that cannot reach them because they are busy or do not answer. This is accomplished in a manner similar to leaving Video-Mail, through the VMDI or communicated with a human video operator.

E. Retrieving Video-Mail and Video On Demand Users have the choice of periodically polling their video-mail for new messages, or having the video-mail server call them periodically when they have a new message waiting.

Configuration is done through the VMDI or human video operator. Managing and checking video-mail is also performed through the VMDI or communicated with a human video operator. A choice of video to view for Video On Demand (VOD) is provided through the VMDI. These videos can be previously recorded video-conferences, training videos, etc. and are stored on the Video Content Engine.

F. Video-conference Scheduling A user can navigate through the VMDI or Internet 10 WWW forms, or communicate with a human video operator to schedule a conference in the Conference Space. The information is stored on the Conference Reservation Engine 8. The other conference participants are notified of the schedule with a video-mail, e-mail message or otherwise. An optional reminder is provided for all registered conference participants at a particular time (e.g. 1 hour before the conference), through video-mail (or e-mail, voice-mail, paging service or any other available notification method).

G. Virtual Reality For multiple party conferences, a virtual meeting place can be generated by the Virtual Reality Space Engine. The implementation of the interface includes an embodiment based on VRML. Each person is in control of an "avatar." Each avatar can have many different features such as visual representation (static representation or live video "head") and audio (voice or music). Data exchange and collaboration are all actions that can be performed in each VR conference room. The private MBONE network allows the multi-casting of conference member's data streams. Since everyone has a different view when interacting in VR-space, the VR Space Engine can optimize the broadcast of everyone's incoming H.263

streams to everyone else by multi-casting only those avatar streams in view for each particular avatar.

XIV. VIDEO-CONFERENCING ARCHITECTURE MCI Video-Conferencing describes an architecture for multimedia communications including real-time voice, video and data or any combination, including video telephony. The architecture also defines inter-operation with other video-conferencing standards. The architecture also defines multipoint configurations and control, directory services and video mail services.

A. Features Video-Conferencing architecture is a multimedia services system and is designed to provide a number of features and functions including, Point-to-Point Video Telephony Multimedia video-conferencing with a MCU for control and multimedia information processing Support for Gateways for interworking with other video-conferencing systems based on ITU H.320 and ITU H.324 standards Support for real-time voice, video and data or any combination Multimedia information streams are transported between the end-user terminals using standard transport protocol RTP Support for dynamic capability exchange and mode preferences, like ITU H.263 video and ITU G.723 audio, between end-user terminals Figure 19C illustrates a Video-Conferencing Architecture in accordance with a preferred embodiment. The components and details of the video-conferencing architecture are detailed below.

B. Components The Video-Conferencing System is comprised of a set of components including, End-User Terminals LAN Interconnect System ITU H.323 Server

Support Service Units 1. End-User Terminals The end-user terminals are the end points of communication. Users communicate and participate in video conferences using the end-user terminals. End-user terminals, including ITU H.323 terminals 1 & 8, ITU H.320 terminal 9 and ITU H.324 terminal 10, are interconnected through the ITU H.323 Server which provides the call control, multi-point control and gateway functions. End-User terminals are capable of multimedia input and output and are equipped with telephone instruments, microphones, video cameras, video display monitors and keyboards.

2. LAN Interconnect System The LAN Interconnect System 3 is the interface system between the MCI Switch Network 2 and the different H.323 Systems including H.323 Server 4, Video Content Engine 5, Video Mail Server 6 and also the H.323 Directory Server 7.

End-User terminals participating in video-telephony sessions or video-conferencing sessions establish communication links with the MCI switch network and communicate with the H.323 Server through the

LAN Interconnect System. The LAN interconnect system provides ACD-like functionality for the H.323 video-conferencing system.

3. ITU H.323 Server The H.323 Server 4 provides a variety of services including call control, multipoint control.

multipoint processing, and gateway services for interworking between terminals supporting different video-conferencing standards like ITU H.320 and ITU H.324.

The H.323 Server is comprised of a set of individual components which communicate with each other and with the other external systems like end-user terminals, video mail server and H.323 directory server. The different components of the H.323 Server include: H.323 Gatekeeper Operator Services Module H.323 Multipoint Control Unit (MCU)

H.323 Gateway 4. Gatekeeper The H.323 Gatekeeper provides call control services to the H.323 terminals and Gateway units. The Gatekeeper provides a variety of services including: Call Control Signaling with terminals, gateways and MCU; Admissions Control for access to the video-conferencing system; Call Authorization; Bandwidth control and management; Transport Address Translation for translating addresses between different kinds of interworking video-conferencing systems; Call Management of on-going calls; Interfaces with the Directory Server[7] to provide directory services; and Interfaces with the Video Mail Server[6] for video mail services.

The Gatekeeper uses the ITU H.225 stream packetization and synchronization procedures for the different services, and is tightly integrated with the Operator Services Module for offering manual operator services.

5. Operator Services Module The Operator Services Module offers manual/automatic operator services and is tightly integrated with the gatekeeper. The manual or the automatic operator terminal, located elsewhere on the LAN, interacts with the gatekeeper through the Operator Services Module to provide all the required operator services.

6. Multipoint Control Unit (MCU) The MCU is comprised of the Multipoint Controller and the Multipoint Processor and together provides multipoint control and processing services for video-conferences. The multipoint controller provides control functions to support conferences between three or more terminals. The multipoint controller carries out capabilities exchange with each terminal in a multipoint conference. The multipoint processor provides for the processing of audio, video

and/or data streams including mixing, switching and other required processing under the control of the multipoint controller. The MCU uses ITU H.245 messages and methods to implement the features and functions of the multipoint controller and the multipoint processor.

7. Gateway The H.323 Gateway provides appropriate translation between the various transmission formats. The translation services include, Call Signaling message translation between H.225 and H.221 which is the part of the H.320 system; Communication procedures translation between H.245 and H.242; and Translation between the video, audio and data formats like H.263, H.261, G.723, G.728 and T.120.

The H.323 Gateway provides conversion functions for transmission format, call setup and control signals and procedures.

8. Support Service Units The Support Service Units include the H.323 Directory Server 7, the Video-Mail Server 6 and the Video Content Engine 5 which interact with the H.323 Server for providing different services to the end-user terminals. The H.323 Directory Server provides directory services and interacts with the gatekeeper unit of the H.323 Server. The Video Mail Server is the repository of all the video mail generated by the H.323 system and interacts with the gatekeeper unit of the H.323 server for the creation and playback of video mail. The Video Content Engine is the repository of all other types of video content which can be served to the end-user terminals. The Video Content Engine interacts with the gatekeeper unit of the H.323 Server.

C. Overview The H.323 based video-conferencing architecture completely describes an architecture for multimedia communications including real-time voice, video and data, or any combination including video telephony. Users with H.323 terminals can participate in a multimedia

video-conferencing session, a point-to-point video telephony session, or an audio only session with other terminal users not equipped with video facilities. The architecture also includes gateways for interworking with other video-conferencing terminals based on standards like ITU H.320 and ITU H.324.

The architecture includes a directory server for offering complete directory services including search facilities. A video mail server is an integral part of the architecture providing for the recording and playback of video mail. A video content engine is also part of the overall architecture for offering multimedia content delivery services.

H.323 terminals participating in a video-conferencing or a video telephony session communicate with the H.323 server through the MCI switch network. The H.323 server offers a variety of services including call control, information stream delivery, multi-point control and also gateway services for interworking with H.320 or H.324 terminals. The server also offers directory services and video mail services.

A H.323 terminal initiating a video call establishes a communication link with the H.323 Server through the MCI switch network. On admission to the network by the H.323 server, the server offers a directory of other available terminals to the call initiating terminal which selects a destination terminal or a destination group to participate in a video conference. The server then sets up a communication link with the selected destination terminal or terminals and finally bridges the calling terminal and the called terminal/terminals. If the destination terminal is unavailable or busy, the server offers the calling terminal an option to deposit a video mail. The server also notifies the recipient of the video mail and offers the recipient services for retrieval of the video mail on-demand. Additional services like content delivery on-demand to H.323 terminals are also offered and controlled by the H.323 server.

D. Call Flow Example The Call Flow for the H.323 architecture based video-conferencing is explained in detail for different call types including, Point-to-Point Calls including calls to other H.323, H.320 and H.324 terminals; and Multipoint Video-Conference Calls.

Figure 19C illustrates various call flows in accordance with a preferred embodiment.

1. Point-to-Point Calls a) Case 1: H.323 Terminal to another H.323 Terminal A call initiating H.323 terminal 1 initiates a call to another H.323 terminal[8] through the MCI Switch Network. The gatekeeper is involved in controlling the session including call establishment and call control. The Terminal end-user interface is any commercially available Web-browser.

Calling terminal 1 initiates a dial-up call to the MCI Switch network; the call is terminated on the H.323 Gatekeeper module of the H.323 Server 4 through the LAN Interconnect 3 system; a PPP link is established between the calling terminal and the Gatekeeper 4 on a well- know unreliable transport address/port; Calling terminal sends a admission request message to the Gatekeeper[4] The Gatekeeper 4 sends an admission confirm message and communicates with the Directory Server 7 and sends back directory information to calling terminal for display at the calling terminal, and the directory information is displayed as a web- page along with a choice of calling modes including Point-to-Point or Conference mode; the admissions exchange is followed by the setting up of a reliable connection for H.225 call control messaging on a well known port; the terminal user chooses the point-to-point mode and also chooses the destination of the call. This is the setup request message; the gatekeeper 4 together with the operator services module/operator proceeds with calling the called terminal 8 with a setup request; if setup request fails, the gatekeeper 4 informs the calling terminal 1 of the failure and provides an option for the calling terminal 1 to leave a video mail; if the user at calling terminal 1 chooses to leave a video mail for user at the destination terminal 8, the gatekeeper 4 establishes a connection with the Video Mail Server 6 and receives a reliable port address from the mail server 6 for a H.245 connection;

the gatekeeper 4 additionally establishes a connection for H.225 call control with the video mail server 6.

the gatekeeper 4 in-turn sends a reliable port address to calling terminal 1 for H.245 control channel. The gatekeeper 4 may be involved in H.245 control channel communications; the calling terminal 1 establishes a reliable connection for H.245 control channel and H.245 procedures like capability exchange, mode preferences etc. are carried out; after the capabilities exchange, H.245 procedures will be used to establish logical channels for the different media streams; the capabilities exchange also involves determination of dynamic port addresses for the transport of the different media streams; the media streams are transported over the dynamic ports in the various logical channels; once the terminal has completed the video mail, it closes the logical channel for video after stopping transmission of the video stream; data transmission is stopped and logical channel for data is closed; audio transmission is stopped and logical channel for audio is closed; H.245 call clearing message is sent to the peer entity; calling terminal 1 transmits a disconnect message on the H.225 port to the gatekeeper 7 which in turn sends the disconnect message to the video mail server 6; the disconnect messages are acknowledged and the call is disconnected; if the setup request is a success, called terminal 8 responds with a connect message which include a reliable port address for H.245 connection; the gatekeeper 4 responds to the calling terminal 1 with the connect message along with the port address for the H.245 control channel communications; calling terminal 1 sets up a connection for H.225 call control signaling with the gateway 4, establishes another connection for H.245 control channel communications and responds to the gateway 4 with connect acknowledgment message; the gatekeeper 4 in-turn sends the connect acknowledgment message to called terminal 8.

called terminal 8 now sets up a H.225 call control connection and also establishes another connection for H.245 with the gatekeeper 4 for control channel communications; the terminals, having established a H.245 control channel for reliable communication, exchange capabilities and other initial procedures of H.245, and an audio channel may be optionally opened before the capabilities exchange; following the capabilities exchange, logical channels over dynamic ports are established for each of the media streams; once the media logical channels are open over dynamic ports, media information can be exchanged; during the session, H.245 control procedures may be invoked for changing the channel structure like mode control, capability, etc., also H.225 control channel is for specific procedures as requested by the gatekeeper[4] including call status, bandwidth allocation, etc.; for termination, either terminal may initiate a stop video message, discontinue video transmission and then close the logical channel for video; data transmission is discontinued and the logical channel for data is closed; audio transmission is discontinued and logical channel for audio is closed; H.245 end session message is sent and transmission on the control channel is stopped and the control channel is closed; terminal receiving the end session message will repeat the closing procedures and then H.225 call signaling channel is used for call clearing; and terminal initiating the termination will send a disconnect message on the H.225 control channel to the gatekeeper 4 which in turn sends a disconnect message to the peer terminal. The peer terminal acknowledges the disconnect which is forwarded to the initiating terminal and the call is finally released.

b) Case 2: H.323 Terminal to H.320 Terminal A call initiated from a H.323 terminal 1 invokes a call to a H.320 terminal 9 through an MCI Switch Network. The gatekeeper along with the gateway is involved in controlling the session including call establishment and call control. A terminal end-user interface is any of the commercially available Web-browsers or a similar interface.

The call flow is similar to a H.323 terminal calling another H.323 terminal as explained in the previous case except that a gateway 4 component is introduced between the gatekeeper 4 and the called terminal 9. The gateway transcodes H.323 messages including audio, video, data and control to H.320 messages and vice-versa. If the H.320 terminal 9 initiates a call to a H.323 terminal[1], the initial dial-up routine is performed by the gateway and then the gatekeeper takes over the call control and the call proceeds as explained in the previous case.

c) Case 3: H.323 Terminal to H.324 Terminal Call initiating H.323 terminal 1 initiates a call to a H.324 terminal 10 through the MCI Switch Network. The gatekeeper along with the gateway is involved in controlling the session including call establishment and call control. The Terminal end-user interface is a Web-browser or a similar interface.

The call flow is similar to a H.323 terminal calling another H.323 terminal as explained in the previous case except that a gateway 4 component is introduced between the gatekeeper 4 and the called terminal 9.

The gateway 4 transcodes H.323 messages including audio, video, data and control to H.324 messages and vice-versa.

If the H.324 terminal 10 initiates a call to a H.323 terminal 1, the initial dial-up routine is performed by the gateway and then the gatekeeper takes over the call control and the call proceeds as explained in the previous case.

2. Multipoint Video-Conference Calls In the case of multipoint video-conference, all the terminals exchange initial call signaling and setup messages with the gatekeeper 4 and then are connected to the Multipoint Controller 4 for the actual conference including H.245 control channel messaging through the gatekeeper 4.

The following are the considerations for setting up a conference: After the initial admission control message exchange, the users are presented with a web page with information about conference type and a dynamic list of participants.

Participants joining later are presented with a web page with conference information and also are requested to enter authentication information All users get connected to the multipoint controller[4] through the gatekeeper[4] The multipoint controller[4] distributes information among the various participants E. Conclusion The video-conferencing architecture is a total solution for multimedia communications including real-time voice, video and data, or any combination, including point-to-point video telephony. The architecture defines interworking with other systems utilizing ITU recommendations.

Additional services including directory services and video mail services are also part of the overall architecture.

XV. VIDEO STORE AND FORWARD ARCHITECTURE The Video Store and Forward Architecture describes a video-on-demand content delivery system. The content may include video and audio or audio only. Input source for the content is from the existing video-conferencing facility of MCI or from any video/ audio source. Input video is stored in a Digital Library in different standard formats like ITU H.320, ITU H.324 ITU H.263 or MPEG and delivered to the clients in the requested format. Delivery is at different speeds to the clients either on the Internet or on dial-up lines including ISDN and with a single storage for each of the different formats.

A. Features The Video Store and Forward Architecture is designed with a rich set of features and functionality including: Delivers Video and Audio on demand:

Supports different compression and transmission standards including ITU H.320, ITU H.324, MPEG and ITU H.263 on both IP (Internet Protocol) and RTP (Real Time Transport Protocol); Supports content delivery on the Internet, by dial-up ISDN lines and by low speed (28.8kbps) Analog Telephone lines; Supports single source of content and multiple storage and delivery formats and multiple delivery speeds; and Supports Content Management and Archival in multiple formats.

B. Architecture Figure 19D is a Video Store and Forward Architecture in accordance with a preferred embodiment.

C. Components The Video Store and Forward architecture can be completely described by the following components.

Content Creation and Transcoding.

Content Management and Delivery.

Content Retrieval and Display.

1. Content Creation and Transcoding Input sources include analog video, video from Multi-Point Control Unit (MCU) and other video sources la and ib. Input content is converted to standard formats like ITU H.261, ITU H.263, ITU H.320, ITU H.263, ITU H.324, MPEG and also formats to support delivery of H.263 over RTP and H.263 over an Internet Protocol 2 and 3. Input can initially be coded as H.263 and optionally transcoded into the various other formats and stored 2. The transcoded content is stored on different servers, one for each content type to serve the various clients each supporting a different format 5a, 5b, 5c, 5d, 5e and 5f.

2. Content Management and Delivery Content is stored on different servers with each server supporting a specific format and is managed by a Digital Library consisting of: - Index Server for managing the indexes and archival of content 4, - Object Servers for storage of content 5a, 5b, 5c, 5d, 5e and 5f, - Proxy Client as a front end to the Index and Object Server and interacting with the different clients requesting for content 6.

Content Delivery is by: - Internet, - Dial-up ISDN lines, - Dial-up Analog Telephone lines at 28.8kbps, and Content format is either a MPEG Stream, H.320 Stream, H.324 Stream, or a H.263 Stream transported over IP or RTP.

3. Content Retrieval and Display Content Retrieval is by clients supporting various formats: - MPEG Client - 7a; - ITU H.263 Client supporting RTP - 7b; - ITU H.263 Client supporting IP - 7c; - ITU H.320 Client - 7d; and - ITU H.324 Client - 7e.

Content is retrieved by the different clients on demand and displayed on a local display.

Clients support VCR like functions like fast-forward, re-wind, etc.

D. Overview Analog Video from different sources and H.320 video from an MCU is received as input and transcoded into various formats as required like ITU H.324, ITU H.261, ITU H.263 or MPEG

and stored on the different Object Servers dedicated for each of the formats. The Object Servers are in turn managed by the Index Server and are together called a Digital Library.

Any request from the clients for content is received by the Index Server and in turn serviced by the Object Server through a Proxy Client.

The Index Server or the Library Server respond to requests from the proxy client and store, update and retrieve objects like H.261, H.263 or MPEG multimedia infomation on the object servers. Then they direct the object server to deliver the retrieved information back to the proxy client. The Index Server has the complete index information of all the different objects stored on the object servers and also information on which of the object server the information is residing on. The index information available on the Index Server is accessible by the proxy client for retrieval of multimedia content from the different object servers.

Security and access control is also part of the index server functionality.

The Object Servers are an integral part of the Digital Library providing physical storage and acting as the repository for the multimedia content, including the video-conferencing information stream from the conferencing facilities. The multimedia content is stored in standard formats which can be retrieved by the proxy client on demand. Each of the Object Servers are dedicated for a specific format of multimedia content like H.261, H.263, MPEG, etc. The organization and index information of the multimedia content including information about the specific object server dedicated for a multimedia format is managed by the index server. The Object Server delivers the stored multimedia content to the proxy client upon receiving specific instructions from the index server.

The Proxy Client is the front end of the digital library and is accessed by all the clients through the Internet for on-demand multimedia content. The Proxy Client also is a World Wide Web (WWW) Server and delivers a page to the clients when accessed. The clients interact with the Proxy Client and thereby with the Digital Library through the WWW pages.

Clients request multimedia content by interacting with the WWW pages. The Proxy Client receives the request from the clients through the WWW pages and processes the request. The Proxy Client then communicates with the index server with object queries as requested by the client. The index server then communicates with one of the object servers dedicated to the requested multimedia format and, based on the index information available at the index

server, directs the object servers to deliver the requested multimedia content to the Proxy Client. The Proxy Client receives the multimedia content from the object server and delivers it to the client making the request.

The Clients connect to the Servers either through the Internet or by dial-up connections on an ISDN line or an Analog line at 28.8 Kbps depending on the video format requested and the client capabilities. A H.320 client connects by an ISDN line and a H.324 client requests services on an analog telephone line at 28.8 Kbps. A MPEG client or a H 263 client using RTP or a H.263 client using IP request services through the Internet. The front-ends for multimedia content query and display like the WWW browsers are integrated as a part of the Client and provide an easy-to-use interface for the end-users.

A request for video from the client is received by the proxy client which routes the request to the Index Server which is turn processes the request and communicates with a specific Object Server in addition to indexing the content for delivery. The Object Server delivers the requested content to the client through the Internet. In the case of the dial-up links, the content is delivered back on the already established link.

In sum, the Video Store and Forward architecture describes a comprehensive system for the creation, transcoding, storage, archiving, management and delivery of video and audio or audio on demand. The delivery of video and audio or audio will be on the Internet or by ISDN or Analog Telephone dial-up lines. Content including video and audio or audio is delivered at various data rates from individual storage locations, each serving a different delivery speed.

XVI. VIDEO OPERATOR A. Hardware Architecture Figure 96 shows the system hardware for allowing a video operator to participate in a video conference or video call, providing numerous services to the video callers. Among the services provided are: answering incoming video calls or dialing out to customer sites; accessing a system for maintaining video conference schedules, joining callers using Bandwidth on Demand Interoperability Group ("BONDING") calls or International

Telecommunication Union-Telecommunication Standardization Sector ("ITU-T") standard H.320 Multi-rate Bearer Service (MRBS) Integrated Services Digital Network ("ISDN") calls into a video conference or video call; monitoring, viewing and recording any video conference or video call; playing back video conferences or video calls recorded earlier; and offering assistance to or responding to inquiries from video conference callers during video conferences or video calls.

The system hardware is comprised of a Video Operator Terminal 40001, a Call Server 40002,

a multimedia hub ("MM Hub") 40003, wide area network hubs ("WAN Hubs") 40004, a multi-point conferencing unit ("MCU") 40005, a BONDING Server 40006, a Client Terminal 40007, and a switching network ("MCI") 40008.

In one embodiment, the Video Operator Terminal 40001 is a Pentium-based personal computer with a processing speed of 90 MHz or greater, 32MB RAM, and a hard disk drive with at least 1.0GB storage space. The operating system in this embodiment is Microsoft's Windows 95. Special features include Incite Multimedia Communications Program ("MCP") software, an H.320 video coder/decoder ("codec") card for audio and video compression (e.g.

Zydacron's Z240 codec), and an isochronous Ethernet ("isoEthernet") network interface card.

Incite's MCP manages the isoEthernet network interface card to create the equivalent of 96 ISDN B-channels in isochronous channels for transmission of video signals.

The Call Server 40002 in this embodiment is a Pentium-based personal computer with a processing speed of 90 MHz or greater, 32MB RAM, and a hard disk drive with at least 1.0GB storage space. The operating system is Microsoft's Windows NT Server. Special features include the Incite Call Server services and an Ethernet network interface card.

Different embodiments of the system accommodate any model of MM Hub 40003 and any model of WAN Hub 40004. In one embodiment, the MM Hub 40003 is the Incite Multimedia Hub, and the WAN Hub is the Incite WAN Hub. The MM Hub 40003 is a local area network ("LAN") hub that connects, via numerous ports supporting isoEthernet interfaces each with a bandwidth consisting of 96 full-duplex B-channels, to personal computers such as the Video Operator Terminal 40001 and the BONDING Server 40006, to WAN Hubs 40004, or to other cascaded MM Hubs. In addition, the MM Hub 40003 can

accept up to ten Mbps of Ethernet data via an Ethernet interface such as the one from the Call Server 40002. The WAN Hub 40004 acts as an interface between an MM Hub 40003 and a public or private switched network such as MCI 40008, enabling video conferencing to extend beyond the WAN or LAN containing the MM Hub 40003 and WAN Hub 40004.

Different embodiments of the system also accommodate various manufacturers' MCU 40005 devices. The function of an MCU 40005 is to allow video conference callers using a variety of different devices, possibly communicating over different circuit-based digital networks, to communicate with one another in a single video conference. For example, one embodiment employs VideoServer's Multimedia Conference Server ("MCS"), which mixes audio to allow any one video conference caller to hear the complete video conference discussion and processes video to allow each video conference caller to see all other callers simultaneously.

In one embodiment, the BONDING Server 40006 is a Pentium-based personal computer with a processing speed of 90 MHz or greater, 32MB RAM, and a hard disk drive with at least 1.OGB storage space. The operating system in this embodiment is Microsoft's Windows 95.

Special features include Incite BONDING Server software, a Digital Signal Processor ("DSP") card (such as Texas Instrument's "TMS320C80" DSP), and an isoEthernet network interface card. Where a Client Terminal 40007 makes BONDING or Aggregated video calls, the BONDING Server 40006 converts the calls to multi-rate ISDN calls used within the video operator platform.

In a preferred embodiment, the Client Terminal a Pentium-based personal computer with a processing speed of 90 MHz or greater, 32MB RAM, and a hard disk drive with at least 1.0GB storage space. The operating system is Microsoft's Windows 95 in this embodiment, and the Client Terminal 40007 is equipped with audio and video equipment making it compatible with ITU-T standard H.320.

In this embodiment, the switching network is an integrated services digital network ("ISDN") provided by MCI 40008.

The Video Operator Terminal 40001 is connected to the MM Hub 40003 via an isoEthernet interface with a bandwidth of 96 full-duplex B-channels, which allows each video operator to

manage up to eight video conferencing clients, each client employing a Client Terminal 40007. The MM Hub 40003 is connected to WAN Hubs 40004 via similar isoEthernet local area network ("LAN") connections. One WAN Hub 40004 connects through MCI 40008 to an MCU 40005 via multi-rate ISDN interfaces. Another WAN Hub 40004 connects to MCI 40008 via a multi-rate ISDN interface, and MCI connects to each Client Terminal 40007 via a BONDING or multi-rate ISDN interface. In a three-way connection, the MCU 40005, the Call Server 40002 and the MM Hub 40003 are connected to one another through an Ethernet wide area network ("WAN") 40009. The MM Hub 40003 is also connected to a BONDING Server 40006 via an isoEthernet interface with a bandwidth of 248 B-channels in full "iso" mode.

B. Video Operator Console Figure 97 shows one embodiment of the system for enabling a video operator to manage video conference calls, which includes a Video Operator Console system 40101 and external systems and interfaces 40108 through 40117.

The Video Operator Console system 40101 is comprised of a Graphical User Interface ("GUI") 40102, a Software System 40103 and a Media Control system 40107. The GUI 40102 interacts with both the Software System 40103 and the Media Control system 40107 to allow a video operator to perform all functions of the video operator invention from the Video Operator Terminal [40001 Figure 96] using the Video Operator Console system 40101.

The Software System 40103 implements the following systems: a Scheduling system 40104 which manages the video operator's schedule; a Recording and Playback system 40105 which records the audio and video input from any call and plays back audio and video input through any call; and a Call System Interface 40106 which acts as an application program interface with the Incite MCP application to manage individual calls by performing swiiching functions such as dial and hold.

The Scheduling system 40104 is connected via an Open Database Connectivity ("ODBC") interface 40108 to a Video Operator Shared Database 40111, which is in turn connected via an interface between VOSD and VRS 40114 to a Videoconference Reservation System ("VRS") 40115. The VRS 40115 submits video conference schedules, conference definitions and site definitions to the Video Operator Shared Database 40111 via the interface 40114 either on a regular basis or on demand by a database agent system within the Video Operator Shared Database 40111. The Video Operator Shared Database 40111, residing in a different computer from that containing the Video Operator Console 40101 in a preferred embodiment, stores all conference and site information such that each Video Operator Console 40101 can retrieve the necessary conference and site configurations for any video conference call. In an alternative embodiment of the external systems associated with the internal Scheduling system 40104, the Video Operator Shared Database 40111 and VRS 40115 may be merged into a single system.

The Recording and Playback system 40105 communicates via a Dynamic Data Exchange ("DDE"), Object Linking and Embedding ("OLE") or Dynamic Link Library ("DLL") interface 40109 with a Video Operator Storage and Playback system 40112 located locally in the Video Operator Terminal [40007 Figure 96]. The Video Operator Storage and Playback system is comprised of a uni-directional recording device 40116 conforming to ITU-T standard H.320 and a uni-directional playback device 40117 conforming to ITU-T standard H.320. Conference calls are recorded by transmitting the digitized audio and video signals from the Video Operator Console 40101 to the H.320 recorder 40116. Conference calls are played back by retrieving a previously recorded conference call from disk storage and transmitting the audio and video signals from the H.320 playback device 40117 to the Video Operator Console.

The Call System Interface system 40106 communicates via a DDE interface 40110 with the Incite MCP application 40113 to manage switching functions such as dial, hold, etc.

The Media Control system 40107 allows the GUI 40102 to communicate directly with external components to manage the GUI 40102 presentation of audio and video. In the embodiment shown in Figure 401, the Media Control system 40107 communicates via a DDE interface 40110 with the Incite MCP application 40113. The Incite MCP application 40113 provides all necessary call setup features and multimedia features such as video window placement and audio control through the DDE interface 40110 to the internal Media Control system 40107, and on to the GUI 40102.

Figure 98 shows a second embodiment of the system for enabling a video operator to manage video conference calls, which includes a Video Operator Console system 40101 and external systems and interfaces 40108 through 40117 and 40203 through 40216. In this embodiment, however, the Software System 40103 is compatible with not only VideoServer's "MCS" 40215 MCU, but also other manufacturers' MCU applications. Thus the internal software system MCU control 40201, the external software system MCU Control System 40208, the MCUs themselves 40214 and 40215, and the interfaces between them 40206, 40210 and 40211, appear in Figure 98. In addition, because not only the Incite MCP 40113 application but also "Other programs with call control interfaces" 40216 may provide necessary call setup and multimedia features in this embodiment, the external Call Control System 40209 is necessary, as are the intervening DDE, OLE or DLL interfaces 40207, 40212 and 40213. This embodiment also includes a Video Store and Forward system 40204 and its DDE, OLE or DLL interface

40203. Finally, the second embodiment adds the internal software system Call Monitor 40202.

As in the first embodiment, the Video Operator Console system 40101 is comprised of a GUI 40102 and a Software System 40103. However, in addition to the Scheduling system 40104, the Recording and Playback system 40105 and the Call System Interface 40106, the software system in the second embodiment includes the MCU control 40201 and the Call Monitor 40202.

The Scheduling system 40104 and associated external systems 40108, 40111, 40114 and 40115 are identical to the those in the first embodiment, pictured in Figure 97 and described above.

The internal MCU control 40201 communicates via a DDE, OLE or DLL interface 40206 with the external MCU Control System 40208 to manage resources and features specific to various different MCU systems. The MCU Control System 40208 communicates either via a ConferenceTalk interface 40211 with the VideoServer MCS 40215 or via another vendor- specific interface 40210 with some Other MCU vendors' MCU 40214.

The Recording and Playback system 40105 communicates via DDE, OLE or DLL interfaces 40109, 40203 with both the Storage and Retrieval system 40205 and the Video Store and Forward system 40204. The Storage and Retrieval system 40205 and Video Store and Forward system 40204 communicate via another DDE, OLE or DLL interface 40207 with the Call Control System 40209. The Call Control System 40209 communicates via another DDE, OLE or DLL interface 40212 with a uni-directional H.320 recorder 40116 and a uni- directional H.320 playback device 40117. Conference calls recorded by transmitting the digitized audio and video signals from the Video Operator Console 40101 through the Storage and Retrieval system 40205 and Call Control System 40209 to the H.320 recorder 40116. Conference calls are played back by retrieving a previously recorded conference call from disk storage and transmitting the audio and video signals from the H.320 playback device 40117 through the Call Control System 40209 and Storage and Retrieval system 40205 to the Video Operator Console 40101. The Video Store and Forward system 40204 operates in a manner similar to the Storage and Retrieval system 40205, communicating between the Recording and Playback system 40105 and the Call Control System 40209.

The call monitor 40202 monitors the state of calls and connections by regularly polling the Call System Interface 40106 within the Video Operator Console Software System 40103.

The Call System Interface 40106 communicates via a DDE, OLE or DLL interface 40207 with the Call Control System 40209 to manage call data, including switching functions such as dial, hold, etc., translating between the Video Operator Console 40101 internal data structures and the Call Control System 40209 data. The Call Control System, in turn, manages either the Incite MCP 40113 or Other programs with call control interfaces 40216.

The Media Control system 40107 communicates via a DDE, OLE or DLL interface with the Call Control System 40209, which communicates via a DDE interface 40110 with the Incite MCP application 40113 or with Other programs with call control interfaces 40216. The Incite MCP application 40113 provides all necessary call setup features and multimedia features such as video window placement and audio control either directly through a DDE interface 40110 to the internal Media Control system 40102 or via the Call Control System 40209. If Other programs with call control interfaces 40216 are used to provide call setup and multimedia features, they communicated with the Media Control system 40107 via the Call Control System 40209.

C. Video Conference Call Flow Figure 98 shows how a video conference call initiated by the video operator is connected through the system pictured in Figure 96. In the first step, illustrated by call flow path 40301, the video operator initiates a call from the Video Operator Terminal 40001 through the MM Hub 40003 to the BONDING Server 40006, where the BONDING Server 40006 converts the call to a BONDING call. In the second step, illustrated by call flow path 40302, the BONDING Server 40006 transmits the BONDING call through the MM Hub 40003 once again, through a WAN Hub 40004, through MCI 40008, and to the Client Terminal 40007.

This step is repeated for each Client Terminal 40007 that will participate in the video conference. In the third step, illustrated by call flow path 40303, the video operator initiates a call from the Video Operator Terminal 40001 through the MM Hub 40003, through a WAN Hub 40004, through MCI 40008, and to the MCU 40005. In the fourth step, illustrated by call flow path 40304, the video operator uses the Video Operator Terminal 40001 to bridge the connections to the Client Terminal 40007 and MCU 40005. Each time the video operator calls a conference call client at its Client Terminal 40007, the MCU's ANI for the particular conference site is passed in the Calling Party Field to identify each client participating in the conference call with the correct conference site. When the MCU is called, the clients' ANI are passed. The MCU can then identify the correct conference site for each call.

In an alternate embodiment, the client initiates a BONDING call from the Client Terminal 40007 through MCI 40005, through a WAN Hub 40004, through the MM Hub 40003, through the BONDING Server 40006, and through the MM Hub 40003 once again to the Video Operator Terminal 40001. The video operator then places a call to the MCU as illustrated in call flow path 40303 and finally bridges the two calls as illustrated in call flow path 40304. To determine the correct conference site for the client-initiated call, the initiating client's ANI is passed to the MCU when the connection is made by the video operator.

While a conference call is in progress, the video operator monitors each of the calls from the Video Operator Terminal 40001. Functions of the video operator include monitoring which

calls remain connected, reconnecting disconnected calls, adding new clients to the conference, or joining the conference to inform the clients regarding conference status.

All calls are disconnected to end a conference, and the video operator shared database [40214 in Figure 98] reflects an updated conference schedule.

D. Video Operator Softwnre System 1. Class Hierarchy Figure 100 shows the class hierarchy for video operator software system classes. In one embodiment using the Visual C++ programming language, the VOObject 40401 class is extended from the Visual C++ base class CObject. VOObject 40401 is a Superclass to all classes of objects in the internal software system for the video operator console system, such that all objects in the internal software system inherit attributes from VOObject 40401.

VOOperator 40402 is an assembly class associated with one VOSchedule 40403 Part-l Class object and one VOUserPreferences 40404 Part-2 Class object, such that exactly one VOSchedule 40403 object and exactly one VOUserPreferences 40404 object are associated with each VOOperator 40402 object. VOSchedule 40403, in turn, is an Assembly Class associated with zero or more VOSchedulable 40405 Party Class objects, such that any number of VOSchedulable 40405 objects may be associated with each VOSchedule 40403 object.

VOSchedulable 40405 is a Superclass to the VOConference 40406 Subclass-l and the VOPlaybackSession 40407 Subclass-2, such that the VOConference 40406 object and the VOPlaybackSession 40407 object inherit attributes from the VOSchedulable 40405 object.

VOConference 40406 is an Assembly Class associated with two or more VOConnection 40412 Party Class objects and zero or one VOPlaybackCall 40415 Part-2 Class objects, such that at least two VOConnection 40412 objects and possibly one VOPlaybackCall 40415 object are associated with each VOConference 40406 object. VOPlaybackSession 40407 is an Assembly Class associated with one VOPlaybackCall 40415 Part-l Class object, such that exactly one VOPlaybackCall 40415 object is associated with each VOPlaybackSession 40407 object.

VOCallObjMgr 40408 is an Assembly Class for zero or more VOCall 40410 Party Class objects, such that any number of VOCall 40410 objects may be associated with each VOCallObjMgr 40408 object. Similarly, VOConnObjMgr 40409 is an Assembly Class for zero or more VOConnection 40412 Part-l Class objects, such that any number of VOConnection 40412 objects may be associated with each VOConnObjMgr 40409 object.

VOConnection 40412 is an Assembly class for two VOCall 40410 Part-l Class objects, such that exactly two VOCall 40410 objects are associated with each VOConnection 40412 object.

VOCall 40410 is a Superclass to the VOPlaybackCall 40415 Subclass-l, such that VOPlaybackCall 40415 objects inherit attributes from the VOCall 40410 object. VOCall 40410 is also an Assembly Class associated with two VOSite 40413 Part-l Class objects, such that exactly two VOSite 40413 objects are associated with each VOCall 40410 object.

Finally, the VOCall 40410 class object uses the VORecorder 40411 class object.

VOSite 40413 is a Superclass to the VOMcuPortSite 40417 Subclass-1, the VOParticipantSite 40418 Subclass-2, and the VOOperatorSite 40419 Subclass-3, such that VOMcuPortSite 40417 objects, VOParticipantSite 40418 objects and VOOperatorSite 40419 objects inherit attributes from the VOSite 40413 object.

VOPlaybackCall 40415 is an Assembly Class associated with one VOMovie 40416, such that exactly one VOMovie 40416 object is associated with each VOPlaybackCall 40415 object.

The VOPlaybackCall 40415 class object also uses the VOPlayer 40414 class object.

VOMessage 40420 object has no associations other than inheriting the attributes of VOObject 40401, the Superclass to all objects in the internal software system.

2. Class and Object details a) VOObject All Internal Software System classes will inherit from the following base class. This base class is extended from the Visual C++ base class Object.

Class VOObject

Base Class CObject Inheritance Type public Friend Classes - (1) Data Types enum senderType~e { SENDER~INTERNAL, SENDER~SCHEDULE, SENDER~CONFERENCE, SENDER~CONNECTION, SENDER~CALL, SENDER TIMER }; enum messageType~e { MSG~DEBUG, MSG~ERROR, MSG~WARNING, MSG APPLICATION~ERROR, MSG ~STATE~UPDATE }; Delivery type flags: DELIVER~MESSAGE~QUEUE, DELIVER LOG FILE, DELIVER~MODAL~DIALOG, DELIVERMODELESS DIALOG, DELIVER~CONSOLEOUTPUT (2) Attributes Access Type Name Description Level static VOOperator* m~pVO video operator pointer static VOSchedule* m~pSchedule scheduler pointer static VOCallObjMgr* mpCallOM Call Object Manager pointer static VOConnectionObjMgr* m pConnOM Connection Object Manager pointer static VOCallSystem* m~pCallSys Call System Interface pointer (3) Methods (a) PostMessage

virtual PostMessage (messageType e type, int errCode, CString info="", int delivery=(DELIVER~MSG QUEUEIDELIVER LOG FILE), senderType senderType=SENDER~INTERNAL, void* sender=NULL); (i) Parameters type The type of message, as defined in the Data Types section errCode The error or warning code as defined in the application's resources.

Info Extra textual information to be passed as part of the message.

delivery Preferred method of message delivery. The delivery options are shown in the Data Types section above. Default method of delivery is stored in the class member variable m~delivery, which should be initialized to both DELIVER~MESSAGE QUEUE and DELIVER LOG FILE only.

senderType The message sender type, as defined in the Data Types section.

Sender A pointer to the object sending the message, i.e. this (ii) Description Use this function to create error, warning, debug, logging and notification messages. It will create a VOMessage object, which will then perform the appropriate actions as specified by the delivery flags.

(b) GetErrorString virtual CString GetErrorString (int errorCode); Return Value: returns a CString object having the error string corresponding to the error code passed.

errorCode parameter: the error code for which you want the error string. Error strings are stored as resources.

This function is called to get a textual description corresponding to an error code.

b) Core Classes (1) Class List Site Participant Site MCU Port Site Video Operator Site Call Playback Call Movie Call Object Manager Connection Connection Object Manager Message Video Operator (2) Class Descriptions (a) Site This is a base class from which classes such as the Participant Site and MCU Port Site classes can be derived from. It's main purpose is to function as a data structure containing pertinent information about who or what is taking part in a Call.

Class VOSite Base Class VOObject Inheritance Type public Friend Classes - (i) Data Types enum Bandwidth~e (MULTIRATE, BONDING, AGGREGATED, HO);

(ii) Attributes Access Type Name Description Level Cstring m~name name of the site ID~t m ID Unique site ID ID~t m~locationID ID for physical location Cstring m~timezone Time zone Cstring m~dialNumber Number(s) to dial. See the Call System Interface section for multiple numbers format.

Bandwidth: moandwidthUsage Bandwidth usage int m~maxNumChannels Maximum number of channels capable VOCall* mqCall pointer to Call object that this Site is apart of.

* Codec or Terminal Type (PictureTel, MCP, etc.) * Call Setup Type (dial-in, dial- out) (b) Participant Site Inherits from VOSite base class.

All customers or conference participants will have their information stored in the VO shared database.

Class VOParticipantSite Base Class VOSite Inheritance Type public Friend Classes -

Attributes Access Type Name Description Level Cstring m~coordinatorName Site coordinator name Cstring m~coordinatorNbr Site coordinator telephone number ID~t m~companyID ID of Company this Site belongs to VOMCUPortSite* m pMCUPort MCU Port Site that is to be associated with in a Connection object (c) MCU Port Site Inherits from VOSite base class.

All conferences take place on an MCU. Each Participant Site needs to connect with a logical "port" on an MCU.

Class VOMcuPortSite Base Class VOSite Inheritance Type public Friend Classes - Attributes Access Type Name Description Level ID~t m mcuID ID to identify the MCU VOParticipantSite* m pParticipant Participant Site that is to be associated with in a

Connection object (d) Video Operator Site Inherits from VOSite base class.

All calls will have the Video Operator Site as one of the sites in a point-to-point call. This structure contains the real ANI of the video operator.

Class VOOperatorSite Base Class VOSite Inheritance Type public Friend Classes - Attributes Access Type Name Description Level ID~t m~operatorID Operator's ID CString m~voicePhone Operator's voice phone number ID~t m~groupID Operator's Group ID ID~t m~superviserID Supervisor's ID CObList m~Calls list of Call objects that this Site is a part of (e) Call A Call is defined as a full duplex H.320 stream between two sites. In all Calls, the Video Operator Site will be one of the sites. A Joined pair of Calls is called a Connection.

Class VOCall Base Class VOObject Inheritance Type public Friend Classes - (i) Data Types enum StateCall~e C ERROR, INACTIVE, INCOMING, DIALING, ACTIVE, DISCONNECTED, HELD, lastCallStates}; enum callOperation~e { ERROR, DIAL, ANSWER, HOLD, PICKUP, DISCONNECT, HANGUP, lastCallOperations } (ii) Attributes Access Type Name Description Level ID~t m~ID call ID VOSite* m~pSite other end of a call site (Participant, MCU Port or unknown) VOOperatorSite* m pOperatorSite Operator site boolean m~operatorInitiated TRUE if the call is initiated by the operator (default) CTime m~startTime the actual time when the call became active boolean m~expectHangup flag that helps determine whether a Hangup is expected or not.

StateCall~e m~state state of the call StateCall~e [nCallStates] m~transitionTable state transition table [nCallOperations] VORecorder* Recorder recorder object for call VOConnection* m~pConnection pointer to Connection object this call

belongs to.

(iii) Methods Disconnection(); is called when the other end of the line hangs up or the line goes dead. The member variable m~expectHangup should be FALSE. Otherwise, the Call Object Manager's Hangup() operation would have been called.

Reset(); resets the call state to an inactive state RecordingStart(); starts recording the H.320 input pipe of the Call.

RecordingStop(); stops the recording of the Call.

setState(callOperation operation); operation parameter: indicates an operation that has been performed which will result in a change of state Operations that affect the state of the Call should call the setState function after the operation has been performed. This function will change the state of the Call by referencing the current state and the operation in the state-transition table. A VOMessage object will be created, with a type of STATUS~UPDATE and sent to the application queue. The GUI and any other component that reads the application queue will therefore be informed of the status update.

69 Playback Call Inherits from VOCall base class.

In this special case of a Call, the Video Operator audio and video output is replaced with the H.320 stream from the playback of a movie by the Video Operator Storage and Playback external system component.

Class VOPlaybackCall Base Class VOCall Inheritance Type public Friend Classes - (i) Attributes Access Type Name Description Level VOMovie* movie the movie object that will be played VOPlayer* Player Player object that performs the playback (ii) Methods PlaybackStart(); starts playback PlaybackStop(); stops playback (g) Movie A Movie is a recording of an H.320 Call. For Phase 1, the Video Operator Storage and Playback System manages files and H.320 data streams for recording and playback of movies, as well as storage and retrieval.

Class VOMovie Base Class VOObject Inhentance Type public Friend Classes - Attributes Access Type Name Description

Level public ID~t m~movieID movie ID public CString m~description movie description (h) Call Object Manager By having a Call Object Manager to perform the construction and destruction of Call objects, a list of all calls on the video operator's machine can be maintained. This includes calls that are not part of any Conference or Playback Sessions, including incoming calls and general purpose dial-out calls. Operations that affect a Call but do not create or destroy it can be performed by the Call object itself.

Class VOCallObjManager Base Class VOObject Inhentance public Type Friend Classes - (i) Attributes Access Type Name Description Level int m~numChannels total number of unused channels int m~numActive total number of active channels CMapStringToOb m~callList list of calls (ii) Methods Dial0;

Dial(VOCall* pCalling); pCalling parameter: If not NULL, this pointer will be used for the Call object. This is necessary when creating or re-using a Call object that is in an inactive or disconnected state.

Dial performs dial out. The number(s) to Dial are in the m pSite Call member structure.

Answer(); Answer(VOCall* pIncoming); pIncoming parameter: If not NULL, this pointer will be used for the Call object. This is necessary when creating or re-using a Call object that is in an inactive or disconnected state.

Answer answers an incoming call.

Hangup(VOCall* pCall); pCall parameter: pointer to the call Hangup hangs up the call pointed to by pCall Hold(VOCall* pCall); pCall parameter: pointer to the call Hold puts the call pointed to on hold.

VOCall* CallCreate(); VOCall* CallCreate creates a Call object.

VOPlaybackCall* PlaybackCallCreate(); VOPlaybackCall* PlaybackCallCreate() creates a Playback Call object.

VOCall* GetCallPtr(ID~t idCall); idCall parameter: call ID VOCall* GetCallPtr gets the pointer to the call object identified by idCall

(i) Connection A Connection is defined as a pair of Call objects that maintain a Join state, and each Call has the Video Operator Site as a common point for the Join to be implemented.

Class VOConnection Base Class VOObject Inheriiance Type public Fnend Classes - (i) Data Types enum StateConnection~e { ERROR, UNJOINED, JOINED, BROKEN, lastConnectionStates }; enum ConnectionOperation~e { ERROR, JOIN, UNJOIN, BREAK, RESET, lastConnectionOperations }; (ii) Attributes Access Type Name Description Level VOCall* m pParticipantCall pointer to the Participant Call VOCall* m pMCUPortCall pointer to the MCU Port Call VOParticipantSite* mqParticipantSite pointer to the Participant Site VOMCUSite* m~pMCUPortSite pointer to the MCU Port Site CTime mjoinTime time ofjoin VOMovie* movie movie pointer for recording or playback boolean m~expectBreak flag that helps determine whether a Break is expected or not.

StateConnection e m~state state of the connection StateConnectione m transitionTable state transition table [nConnectionStates] [nConnectionOps] VOConference* mqConference pointer to the Conference that this Connection is a part of.

(iii) Methods Join(); joins the Participant and MCU Port Calls.

Unjoin(); unjoins the Participant and MCU Port Calls.

SetP articipantCall(VOCall * participantCall); participantCall parameter: pointer to a Call object. SetParticipantCall sets the Call to be the Participant Call. This is useful when managing unknown incoming calls or for last minute participant substitution.

SetMCUPortCall(VOCall* mcuPortCall); mcuPortCall parameter: pointer to a Call SetMCUPortCall sets the Call to be the MCU Port Call. This is useful when managing unknown incoming calls or for last minute call site substitution.

DoParticipantCall(); calls the Participant Site and sets it as the Participant Call.

DoMCUPortCall(); calls the MCU Port Site and sets it as the MCU Port Call.

setState (ConnectionOperation operation); operation parameter: the operation that has been performed which will result in a change of state.

Operations that affect the state of the Connection should call the setState function after the operation has been performed. This function will change the state of the Connection by referencing the current state and the operation in the state-transition table. A VOMessage object will be created, with a type of STATUS~UPDATE and sent to the application queue.

The GUI and any other component that reads the application queue will therefore be informed of the status update.

protected Break(); is called when a Joined Connection becomes Un-joined. If the member variable m~expectBreak is FALSE then one of the Calls must have unexpectedly been disconnected. Otherwise, the Connection's Unjoin() operation would have been called.

protected Reset (); resets the state of the Connection to UNJOINED.

(j) Connection Object Manager Similarly with the Call Object Manager, a list of all Connections in operation on the video operator's machine must be maintained. All operations that result in the creation or deletion of a Connection must use the Connection Object Manager.

Class VOConnectionObjMgr Base Class VOObject Inheritance Type public Friend Classes - (i) Attributes

Access Type Name Description Level CMapStringToOb m~connectionsList list of all connections int m~numJoined number ofjoined connections (ii) Methods VOConnection* Create(); Return Value: pointer to Connection object VOConnection* Create creates a new Connection object and adds it to the list.

Remove (VOConnection* oldConnection); oldConnection parameter: connection object to be removed Return Value: returns TRUE if operation successful.

Remove deletes a Connection object and removes it from the list.

VOConnection* GetConnectionPtr(t idConnection); Return Value: a pointer to the connection object idConnection parameter: ID of the Connection VOConnection* GetConnectionPtr returns the pointer to a Connection object identified by its ID.

(k) Message All one-way communication from the Internal System Software to the rest of the Video Operator application, i.e. the Graphical User interface, is sent as messages that get placed on

the Application Queue. The function to create and post a Message is in the base class VOObject, which all Internal System Software classes inherit from. All run-time errors or debugging information is put into a Message object, and posted to the application queue so that an appropriate object will process it according to its type and severity. Therefore all class functions that do not return a specific type will post a Message if something goes wrong, e.g.

out of memory, or debugging information to be displayed by the GUI or logged to a file.

Class VOMessage Base Class VOObject Inheritance Type public Friend Classes - (i) Data Types enum senderType~e { INTERNAL, SCHEDULE, CONFERENCE, CONNECTION, CALL, TIMER }; enum messageType~e { DEBUG, ERROR, WARNING, APPLICATION~ERROR, STATE~UPDATE }; Delivery type flags: DELIVER MESSAGE QUEUE, DELIVER~LOG~FILE, DELIVER~MODAL~DIALOG, DELIVER~MODELESS~DIALOG, DELIVER CONSOLEOUTPUT (ii) Attributes Access Type Name Description Level int m~errorCode error code int m~delivery flags for preferred message delivery when posting.

senderType e m~senderType sender type VOObject* mqObject pointer to the sender messageType e m~messageType type of the message CString m~info message info * priority of message or error * severity of message or error (iii) Methods Post(); posts a message to the application message queue private static AppendLog(); Return Value: returns TRUE if the operation is successful.

This method is called by VOObject::

(l) Video Operator Generally there v
has a Schedule, and a list of custom
Connection Object Manager are als

Class VOOperator Base Class VOC
Type Name Description Level ID~t r

VOSchedule m~schedule schedule
CObList m~operatorSites Operator'